# IRE Transactions on ELECTRONIC COMPUTERS

**Volume EC-7** **SEPTEMBER, 1958** **Number 3**

*Published Quarterly*


## TABLE OF CONTENTS






PUBLISHED BY THE
Professional Group on ELECTRONIC COMPUTERS

**IRE Transactions® on Electronic Computers**

Published by the Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79th Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Price per copy: IRE-PGEC members, $1.00; IRE members, $1.50, nonmembers, $3.00. Yearly subscriptions rate: nonmembers, $17.00; colleges and public libraries, $12.75. Address requests to The Institute of Radio Engineers, 1 East 79th Street, N.Y. 21, N.Y.

*Notice to Authors:* Address all papers and editorial correspondence to J. P. Nash, Missile Systems Division, Lockheed Aircraft Corp., 3251 Hanover Street, Palo Alto, Calif. To avoid delay, 3 copies of papers and figures should be submitted, together with the originals of the figures which will be returned on request. All material will be returned if a paper is not accepted.

# The Chairman's Column

To Members of the PGEC

This year I hope to keep all of you informed on current and future happenings in the PGEC. I hope to do it principally through this column, which will appear (when there is something to be said) in each issue of these Transactions.

On April 29, C. W. Rosenthal of the Administrative Committee attended, in my behalf, a meeting in New York of the national Professional Groups Committee. The following summary is based on his report to me.

1) Joint PG chapters will be approved where necessary and advisable.
2) PG members who are 4 months delinquent in dues will be dropped from membership.
3) There is activity under way to standardize the covers of the various Group Transactions and also the forms used in reviewing submitted papers.
4) The financial status of national Professional Groups was discussed. An Ad Hoc Committee has been formed to study this problem and member's suggestions to cure it (*e.g.*, to accept advertising in the Transactions). Dr. R. M. Emberson, Dr. R. L. McFarlan, Dr. L. C. Van Atta and the undersigned have been appointed to this committee. The chairman will be Dr. John T. Henderson, past president of the IRE and chairman of the board of directors.
5) A survey of the financial affairs of Professional Group chapters indicated that the principal problems were not financial, but rather in personnel and organizational matters. The personnel problem is one of strong leadership and an adequate carry-through from year to year, while organizational difficulties are best helped by the Professional Groups Chapter Coordinator of the cognizant Section.
6) There is some feeling among Professional Groups which do not hold any national meetings of their own, that they should get preferred treatment and time at the national IRE Convention, and that Groups which hold national meetings should restrict their national convention papers to tutorial ones or ones of broad interest; specialized papers should be presented at group national meetings.

During the 1958 Western Joint Computer Conference the Administrative Committee held an informal meeting. Since there had been a meeting about six weeks previously in New York there was no pressing business to transact and consequently, no formal actions were taken.

However, a number of items of business were discussed; significant findings are outlined here.

1) With respect to the problem of obtaining an abstract service to replace one previously conducted by Dr. Harry Huskey on a volunteer basis, the consensus of those present was that the PGEC should spearhead this effort and not wait for the cooperation of other technical societies. Frank Heart of Lincoln Laboratories and Larry Jones of Westinghouse-Baltimore, the Ad Hoc Committee for this problem, are preparing a specific proposal for consideration by the Administrative Committee meeting at its August 21 meeting during WESCON in Los Angeles.
2) Members present felt that we have no need at the moment of financial return from advertising in these Transactions; for the moment, therefore, we will not accept advertising.
3) The National Joint Computer Conference recently amended its proposed charter to meet the objections of the PGEC Administrative Committee. The matter will be voted upon by the Administrative Committee shortly.
4) Those present were in favor of rerunning the membership survey which was conducted in the fall of 1956 since it was felt that significant changes have occurred in the last 18 months. As before, to protect the anonymity of the information, a large public accounting firm will be used as the mail-drop, and will carefully dissociate any identifying information. Since this matter is subject to the vote of the Administrative Committee, no action has been taken yet.
5) The problem of PGEC participation in an International Federation of Computer Societies was discussed. This is a matter which requires IRE attention, and Dr. Arnold Cohen is investigating.
6) As of May 1, the PGEC had 7338 members.
7) Chairman Melville of the Student Activities Committee indicated that he is not in favor of continuing our postdoctoral fellowship beyond this year, primarily because fellowship money is readily available elsewhere, and the PGEC does not receive a significant amount of publicity. An alternate plan will be submitted by Melville for Administrative Committee consideration.
8) Editor Nash indicated that he will recommend some changes in the membership and structure of the editorial board and Publications Committee.

Plans are now firm for an International Conference on Information Processing to be held June 13–21, 1959. Sponsored by UNESCO, the conference will be in Paris. Only invited papers will be used by the U. S. Committee for the ICIP. Further information may be obtained from the committee at Box 4999, Washington 8, D. C. All travel arrangements must be coordinated through this committee.

Willis H. Ware
*Chairman*

# Willis H. Ware

Willis H. Ware (A'43–SM'49) was born in Atlantic City, N. J., on August 31, 1920. He received the B.S. degree from the University of Pennsylvania in 1941, and the S.M. degree in 1942 from Massachusetts Institute of Technology, where he was a Tau Beta Pi Fellow.

From 1942 to 1946 he was employed by the Hazeltine Electronics Corporation for research and development in radar and IFF. In 1946 he became one of the original members of the staff of the Electronic Computer Project at the Institute for Advanced Study, Princeton, N. J. There he worked on the design and development of the large scale general purpose electronic digital computer which was later to set the pattern for the construction of several other "Princeton class machines." At the same time he continued his graduate work at Princeton University, from which he received the Ph.D. degree in 1951. In 1952, Dr. Ware joined the RAND Corporation, where he is continuing his work with digital machines.

Dr. Ware has held several national and local offices in the PGEC. He is National Chairman for 1958–1959 and has held posts as National Vice-Chairman, 1957–1958; National Secretary-Treasurer, 1954–1955; National Chairman of the Student Activities Committee, 1956–1957; Chairman of the Los Angeles Chapter, 1954–1955; Treasurer of the Los Angeles Chapter, 1953–1954. He also has been a member of IRE Technical Committees 8.4 and 8.5 (computer definitions) from 1948 to the present time, and served as Chairman of Committee 8.5 from 1951 to 1958.

Conference Chairman of the 1958 Western Joint Computer Conference, Dr. Ware is currently serving as a member of the IRE national Ad Hoc Committee to review Professional Group Finances. He is also a member of the Technical Program Committee of the International Conference on Information Processing. In 1957 he was Finance Chairman of the Western Joint Computer Conference and also the Professional Groups Coordinator of the Los Angeles section. In 1956 he was the alternate Chairman of the Technical Program Committee of WESCON.

Dr. Ware is a member of Tau Beta Pi, Sigma Xi, Eta Kappa Nu, Pi Mu Epsilon, and the American Association for the Advancement of Science.

# Design of AC Computing Amplifiers Using Transistors*

C. A. KRAUSE† AND R. R. LOWE†

*Summary*—A design philosophy for transistorized analog computing amplifiers is presented. A design procedure for a summing amplifier to drive a specific resolver in a 400 cps system is described, and the performance of the resulting circuit is evaluated. Whereas experience with vacuum tube amplifiers in similar applications has led to the conclusion that the amplifier input impedance should be as large as possible, the inverse is true in the transistor amplifier.

## I. Introduction

IT HAS been felt that the transistor had inherent disadvantages in analog summing amplifiers because of its relatively low input impedance. Investigation shows that this is not the case and that, in general, a current amplifying device is most valuable when its input impedance approaches zero.[1] This and subsequent conclusions resulted from the development of a silicon transistor summing amplifier to drive a resolver of 15 kilohms tuned resistance at a computing frequency of 400 cps. It was desired that the summing resistors be as large as 500 kilohms (this will be seen to have a direct bearing upon the required current gain), and that the computing error due to finite amplifier gain not exceed 0.05 per cent when the closed loop gain is unity. The various configurations in which the amplifier-resolver combination are to be used are shown in Fig. 1. Parallel and series compensator feedback provide multiplication of the input by the sine and/or cosine of the resolver angle, series feedback having potentially higher loop gain but restricted to a single input. Parallel rotor feedback provides for division by the same trigonometric functions. Since the rotor feedback case can never have any more loop gain than the compensator feedback cases, it is the less critical application (ignoring resolver idiosyncrasies). The approach taken was to provide adequate loop gain in the parallel compensator feedback case, so that the series feedback configuration had reduced significance. Thus provision for adequate loop gain and stability margins in the case of parallel compensator feedback is essentially a complete solution.

## II. Summing Amplifier Input Impedance

The impedance at the summing node of a good computing amplifier is very nearly zero. The input impedance seen from a summing input to the amplifier is, then, essentially the value of the summing resistor,

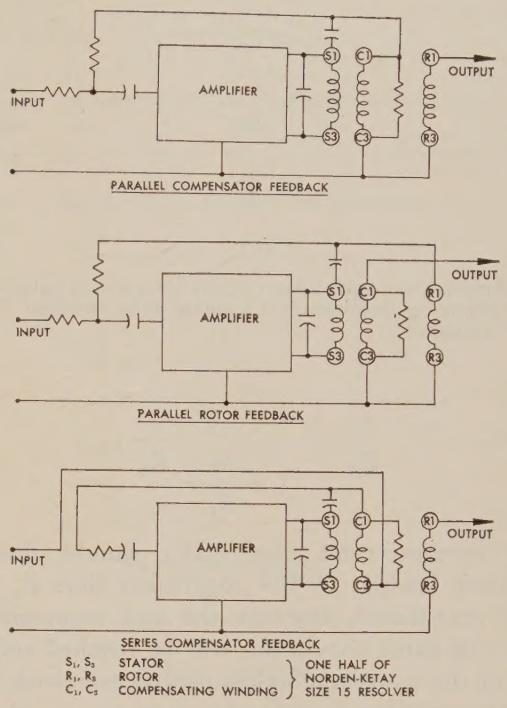



Fig. 1—Various configurations of the amplifier-resolver combination.

regardless of the amplifier's own internal input impedance. In a vacuum tube summing amplifier, the internal input impedance should be as high as possible to prevent deterioration of loop voltage gain by its shunting effect. In a transistor summing amplifier, the internal input impedance should be as small as possible so that a *voltage* error in the relationships at the extremities of the feedback and summing resistors will produce the maximum possible current signal into the transistor. Consider the amplifiers of Fig. 2. In each circuit, the finite gain is evidenced by the fact that $e_i/R_s$ is not equal to $e_0/R_f$, *i.e.*, a non-zero current flows into the summing point. Assuming that the voltage at the amplifier input is small in each case, the error term can be approximated as

$$E = e_i \frac{R_f}{R_s} + e_0.$$

Expressing $E$ in terms of each of the two circuits,

$$\frac{e_g}{E_{vt}} = \frac{1}{1 + \frac{R_f}{R_s} + \frac{R_f}{R_g}},$$

* Manuscript received by the PGEC, December 18, 1956; revised manuscript received, April 18, 1958.

† Norden Division, United Aircraft Corp., Gardena, Calif.

[1] R. L. Wallace, Jr., and G. Reisbeck, "Duality as a guide in transistor circuit design," *Bell Sys. Tech. J.*, vol. 30, pp. 381–418; April, 1951.

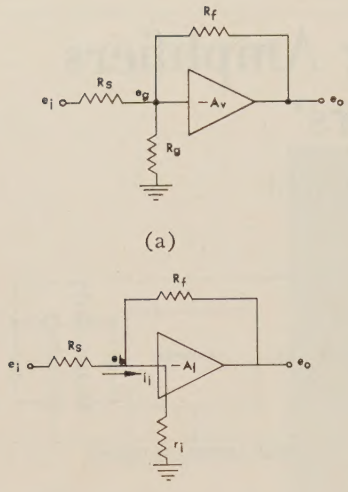



Fig. 2—Approximate equivalent circuits for vacuum tube and transistor summing amplifiers. (a) Vacuum tube amplifier. (b) Transistor amplifier.

and

$$\frac{e_b}{E_{tr}} = \frac{1}{1 + \frac{R_f}{R_s} + \frac{R_f}{r_i}},$$

In the vacuum tube case, $e_g/E_{vt}$ should clearly be maximized, leading to the conclusion that $R_g$ should also be maximized. Because the two expressions are similar, the same conclusion will be reached regarding the $R_1$ of the transistor unless care is exercised. As the transistor amplifies $i_i$ and not $e_b$, and $e_b = i_i r_1$, it can be shown that

$$\frac{i_i}{E_{tr}} = \frac{1}{r_1\left(1 + \frac{R_f}{R_s}\right) R_f}.$$

Thus $r_i$ should be minimized in order to maximize $i_i/E_{tr}$.

Since this conclusion is directly opposed to the vacuum tube case and to some earlier opinions regarding transistor amplifiers, it must be justified. Consider the approximate equivalent circuit of a summing amplifier shown in Fig. 3. Assuming that $e$ is very small for the purpose of computing the power in $R_f$, and defining the power gain, $A_p$, as

$$A_p \cong \frac{e_0^2}{R_L'} \div \frac{e_i^2}{r_i},$$

where

$$R_L' = \frac{R_L R_f}{R_L + R_f},$$

it can be shown that

$$\frac{e_0}{e_s} = \frac{R_f}{R_s} \cdot \frac{1}{1 - \frac{R_f/r_i + R_f/R_s + 1}{\sqrt{A_p R_L / r_i}}}.$$

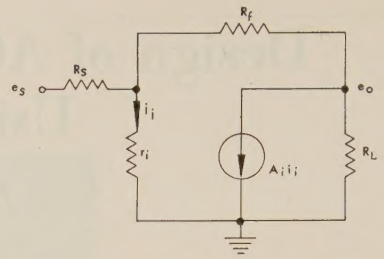



Fig. 3—Approximate equivalent circuit of a transistor summing amplifier. (*Note:* For small-signal, low-frequency application, any degree of rigor may be obtained from this circuit by properly defining $r_i$ and $A_i$.)

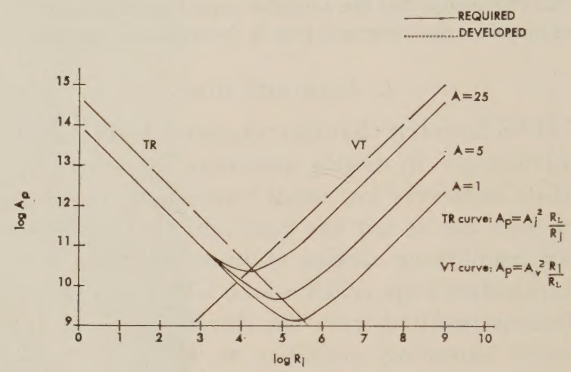



Fig. 4—Required and developed power gain as a function of input resistance for vacuum tube and transistor summing amplifiers.

If the desired ratio between $e_0$ and $e_s$ is $-R_f/R_s$, the magnitude of the error term, $E$, is

$$|E| = \sqrt{\frac{r_i}{A_p R_L}} \Big/ \left(\frac{R_f}{r_i} + \frac{R_f}{R_s} + 1\right).$$

Fig. 4 shows $A_p$ as a function of $r_i$ for various values of $R_f/R_s$ (the desired closed loop gain) with $|E| = 0.05$ per cent.

Referring to the $A_p$-$r_i$ plot, the dashed lines labeled *TR* and *VT* show the developed power gain input impedance for a transistor amplifier and a vacuum tube amplifier, respectively, both of which were designed for the "matched" condition with 25 computing inputs. For the vacuum tube case, an otherwise identical amplifier develops more power gain as the input impedance (grid resistor) is increased, even providing slightly more gain than required for the specified accuracy. This is intuitively satisfying and indicates decreasing power requirement in the input circuit (at constant output) as the grid resistor is increased. For the transistor case, an otherwise identical amplifier develops more power gain as the input impedance (characteristic of the first stage, or intentionally added series resistor) is decreased, again providing slightly more gain than is required for the specified accuracy, and decreased input power requirement at constant output.

Another inference from the $A_p$-$R_1$ plot is that transistor amplifier computing accuracy is relatively in-

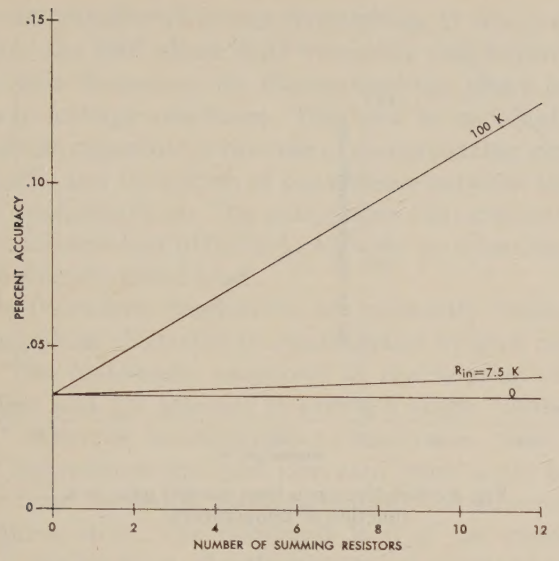



Fig. 5—Accuracy as a function of the number of summing resistors for amplifier current gain of 100,000 and different values of input resistance.

dependent of the number of inputs. To a first approximation, the number of inputs (at equal summing resistance) required to reduce the amplifier accuracy to one-half of the single-input value is equal to the ratio of the summing resistance to the amplifier input resistance. For example, an amplifier having 0.05 per cent accuracy with a 500K feedback resistor and one 500K summing resistor, and with an input resistance of 10K, will have an accuracy of about 0.1 per cent with fifty 500K inputs. Alternatively, multiplication of a single input by variation of an input summing resistor can be carried out with considerably more accuracy than in the comparable vacuum-tube circumstance. The amplifier having characteristics described above can maintain half the single-input accuracy in multiplications up to 50:1. This point is shown graphically in Fig. 5, which shows the accuracy of such an amplifier having a current gain of 100,000 for various values of $R_{in}$. The curve plotted for an $R_{in}$ of 7.5 kilohms is of special significance, since this corresponds approximately to the final design.

## III. Loop Gain

It has been shown that multiple summing resistors have little effect upon a properly designed transistor amplifier. It is also true that the magnitude of the load resistor has little effect upon the vacuum tube amplifier, but a large effect upon the transistor version. As $r_i$ approaches zero,

$$\left.\frac{e_0}{e_i}\right]_{R_1 \to 0} = -\frac{R_f}{R_s} \cdot \frac{A_i}{A_i + \dfrac{R_f + R_L}{R_L}}.$$

Thus a larger value of current gain is required if $R_L$ is reduced. For instance, if the tuned resistance of the resolver is reduced from 15 kilohms to 10 kilohms, the gain requirement is increased almost 50 per cent. The percentage of the amplifier current which is fed back

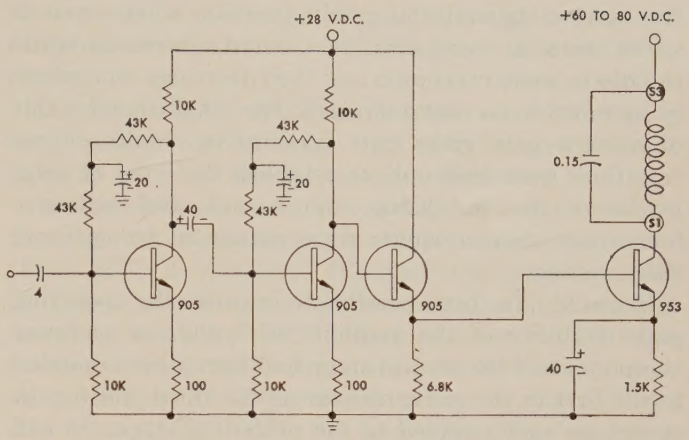
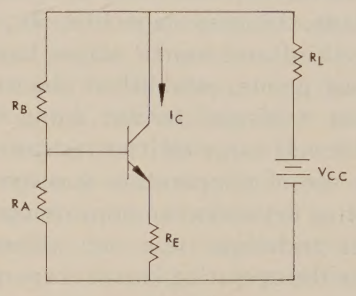



Fig. 6—Final amplifier configuration.




$$I_C = \frac{BI_{CO} + BV_{CC}\left(\frac{1}{r_C} + \frac{1}{R_1 + R_2}\right)}{1 + B\left(\frac{R_1}{R_1 + R_2} + \frac{R_L + R_E + r_e}{r_C}\right)}$$

$$\text{where } R_1 = R_L + (R_E + r_e)\left(\frac{1 + R_B + R_L}{R_A}\right)$$

$$R_2 = R_F + r_b\left(\frac{1 + R_B + R_L}{R_A}\right)$$

Fig. 7—Equations for collector current stabilization. $B$ is the dc current gain at the operating point and $r_b$, $r_e$, $r_c$, and $I_{co}$ are used in the conventional sense.

is approximately inversely proportional to the ratio of the feedback resistor to the load resistor.

## IV. Amplifier Configuration

### *A. Operating Point Stabilization*

The methods of operating point stabilization should be established before the basic amplifier configuration is finalized, since such stabilization can be achieved only at the expense of dynamic current gain. Thus the degree of required stabilization affects the choice of transistor types and the manner in which these transistors must be connected and interconnected. The first two stages of Fig. 6 have been stabilized by means of the collector-to-base feedback resistors, the base-to-ground shunt resistor, and the emitter resistors.

Equations can be developed[2] which express the degree of operating point stability as a function of transistor parameters and external resistors (see Fig. 7). These equations give quite realistic results in the case of germanium transistors, but the current-amplifying properties peculiar to silicon reduce their usefulness. Since

[2] A. W. Lo, R. O. Endres, *et al.*, "Transistor Electronics," Prentice-Hall, Inc., New York, N. Y., pp. 131–154; 1955.

the current gain of the grown junction silicon unit is about zero at zero emitter current, increases quite rapidly to some maximum and then decreases more slowly as emitter current increases, the small-signal value of current gain gives only qualitative results. These equations were used only to establish the order of magnitude of the stabilizing components, and extensive laboratory measurements were employed to optimize their values.

It would have been possible to improve the operating point stability of the amplifier with the use of fewer components if the second stage had been direct-coupled to the first in the same manner as the third and fourth stages are each coupled to the preceding stage. As will be shown in the discussion of low-frequency stabilization, this can be accomplished only at the expense of an undesirably large by-pass capacitor. In the existing amplifier, the third and fourth stages have extremely stable operating points, established almost entirely by the base input voltages derived from the previous stage, and relatively large emitter resistors.

The proper use of temperature-sensitive resistors in the compensating networks can improve operating point stability. This technique was not necessary in this amplifier, since the operating currents are quite narrowly confined in the regions of maximum current gain by the existing networks. No change of maximum undistorted output with temperature was observed, and Fig. 8 shows that the gain decrease with temperature is no more than expected due to current gain decrease with temperature, *i.e.*, current level shift has not contributed significantly to the effect. As will be discussed later, temperature sensitive resistors might be properly incorporated to partially compensate for this gain decrease. The fact that the gain of the amplifier did not significantly increase at high temperatures where the individual transistor current gains have increased 20 per cent or more is due to increased insertion losses caused by higher input and lower output impedances of the individual stages.

### *B. Individual Stages*

The method of connecting the individual stages is largely determined by the desire for maximum current gain from a given number of transistors without the use of transformers. The common base connection is immediately eliminated because it has a current gain of less than unity. The common collector connection is generally less efficient than the common emitter because of the large insertion loss caused by its relatively large input impedance and low output impedance. However, where four stages must produce a net reversal of phase, the use of a single non-phase-reversing common collector makes it unnecessary to obtain a phase reversal in the resolver load, thereby minimizing the adverse affects of interwinding capacities. Since the input impedance of a common collector stage is minimized when its load resistance is smallest, and since the input resistance of the Type 953 transistor is significantly less than the Type 905, making the third stage the single common collector stage minimizes the insertion loss.

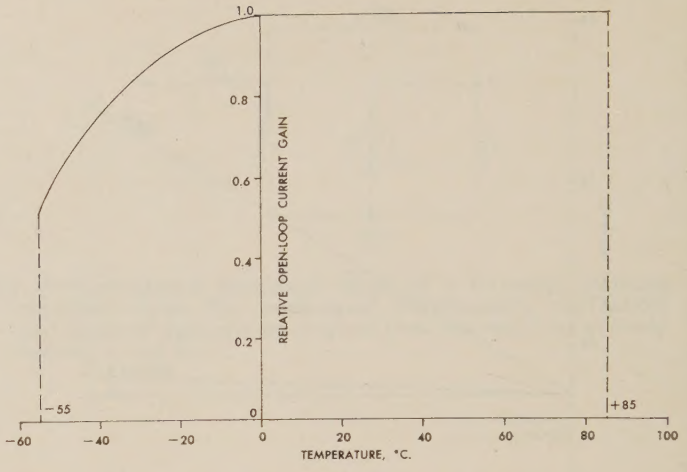



Fig. 8—Relative open loop current gain as a function of temperature.

### *C. Transistor Types*

To the knowledge of the authors, the single manufacturer of production quantities of high quality silicon junction transistors at the time this amplifier was designed was Texas Instruments, Incorporated. Since the required output voltage was to be a minimum of 20 volts rms, only the Types 952 (80 v) and 953 (120 v) have adequate collector voltage maxima. The 953 was chosen because collector voltage maxima are reduced at high temperatures, because an appreciable voltage must be maintained across the transistor for class A operation, because the maximum undistorted output was desirable, and because the price and availability of the two types were identical.

Since the minimum guaranteed common-emitter current gain of the Type 953 is less than 10, and since the required total current gain must be about 100,000, it is desirable that the other three stages have a large guaranteed minimum current gain. This is also desirable from the standpoints of allowing maximum ac and dc degeneration and optimizing low temperature performance. Choice of the Type 905 with a minimum common emitter current gain of about 40 (this is the best available) gives performance, under the most adverse conditions, that is marginally adequate. The current gain thus achieved and its effect upon computing accuracy will be discussed in Section VI.

## V. Dynamic Stabilization

The dynamic stabilization problem can logically be divided into two parts—*e.g.*, instabilities above and below the nominal 400 cps operating frequency. The high-frequency problem is of little concern in this discussion since its origin and its solution both stem from the specific resolver load. Because the particular resolver was tuned and because its dissipation is low, the gain

decreases rapidly with high frequencies. It was possible to place the 180° phase shift frequency well beyond the unity gain frequency by eliminating the effect of the resolver leakage reactance. This can be accomplished by a single capacitor in the case of compensating winding feedback, and by a form of parallel tee network for the other configurations. The gain-phase characteristics of the transistors had little to do with the problem because of the sharply tuned load.

Low-frequency oscillations are primarily caused by leading phase shifts due to coupling and by-pass capacitors. The philosophy employed in the solution of this problem was the attempt to create a single "dominant lead," wherein one resistance-capacitance time constant has reduced the open loop gain below unity before other time constants make significant contributions to the phase shift. The dominant lead of the circuit of Fig. 7 results from the 40 microfarad emitter by-pass capacitor of the fourth stage, the input resistance and current gain of this stage, and the output resistance of the third stage. This time constant, about 0.001 second, is at least an order of magnitude smaller than all others.

If the second stage had been direct-coupled to the first, the by-pass capacitor for the emitter resistor would have been much greater than the 40 microfarad coupling capacitor, as impedances from emitter to ground are multiplied by the stage current gain in their effect upon input impedance. If the large time constant created by the 40 microfarad coupling capacitor were to be maintained, the by-pass capacitor required for direct coupling would have to be about 2000 microfarads, regardless of the size of the emitter resistor.

## VI. Design Modifications for Improved Accuracy

The computing accuracy of the described type of summing amplifier can be improved by several methods. Such methods as reducing the emitter degeneration and/or reducing the size of the summing resistors have the disadvantage of reducing the gain margins. Emitter resistors with a positive temperature coefficient might be used to give the same high-temperature margins with reduced low-temperature degeneration. The computing accuracy might also be improved by obtaining the transistors to a more limited current gain tolerance, but there are inherent disadvantages of selection techniques.

It may also be practical in some cases to employ temperature control such that the lowered current gains at very low temperature are eliminated.

In this particular case, it was originally assumed that the ambient temperature could be maintained at or above 25°C, if necessary. The resultant amplifier under the most adverse conditions maintained the desired accuracy to about −30°C. Reduction of the 100 ohm emitter resistors to 47 ohms, or of the 500 kilohm summing resistors to 300 kilohms maintains the 0.05 per cent accuracy to −55°C, but lowers the gain margins by about 4 db. A minimum "alpha" specification for the type 953 transistor of about 0.94, rather than 0.90, would allow the amplifier to operate within the specified accuracy throughout the temperature range.

An excellent method for improving the performance of such an amplifier, where feasible, is the addition of another transistor stage. This would not only achieve increased computing accuracy, but also allow greater degeneration within the amplifier. The increased degeneration would increase the gain margins by reducing gain dependence on transistor characteristics.

## VII. Conclusion

As originally conceived, the purpose of the amplifier described was to duplicate the characteristics and performance of previous vacuum-tube devices. Early in the course of the preliminary design, fundamental considerations of transistor operation precipitated an approach which resulted in an amplifier which was extremely satisfactory in external performance, but which had internal characteristics quite different from those initially envisioned. Some preconceived notions about input impedance and impedance matching were discarded, and some natural advantages of the transistor in an ac computing amplifier application were discovered. Methods of division can be devised which do not suffer severely from loss of loop gain, and very large numbers of summing inputs are practical. The outstanding disadvantage is that loop gain varies directly with load resistance.

# A Note on Contact Networks for Switching Functions of Four Variables*

RODERICK GOULD†

*Summary*—Several corrections are given to a recent tabulation of two-terminal contact networks realizing the switching functions of four variables. Nineteen new networks which are more economical in contacts than those previously tabulated are also presented and certain four-variable functions possessing a useful complementary relationship are listed. In conclusion this paper indicates an area where further work on four-variable contact network synthesis is needed.

A SWITCHING function of four binary variables is definied by mapping the values 0 and 1 onto the $2^4$ configurations of values of the arguments. It follows that there are $2^{2^4}$, or 65,536, different functions of four variables. These may be divided into 402 classes such that all of the functions of each class are equivalent under argument transformations (*i.e.*, complementation or rearrangement of the independent variables). A list of 402 functions, comprising one representative from each of the 402 classes, was published by Aiken and his Harvard associates in 1951 [1]. The utility of this list is demonstrated by the fact that a vacuum-tube switching circuit for any four-variable function may be obtained from the table of 402 circuits given in [1]. The expression "function of four variables" in this paper refers only to those included in the Harvard list.

In a recent book [2], Higgonet and Gréa included a table of two-terminal contact network realizations for 238 of the 402 four-variable functions; their networks were designed to use as few contacts as possible. A check of this table has revealed a number of errors. The circuits listed for the following functions[1] are incorrect: 49(5-19), 68(6-11), 83(6-26), 84(6-27), 158(7-51), 185(8-22), 211(8-48), 231(8-68). Correct networks for these functions are given in Fig. 1. These new networks were obtained by a synthesis method, described by Gould [3], which involves using the given switching function to obtain a vector space; the latter in turn determines the desired network.

In addition to the errors already mentioned, the following misprints occur in circuits of Higgonet and Gréa [2]: 78(6-21)—at bottom of network, $z'$ should be $z$; 96(6-39)—the unmarked contact connected between $x$ and $z'$ should be labeled $y$; 107(6-50)—in second $x$ contact down from top, $x$ should be $x'$; 136(7-29)—at top of network, $x$ should be $x'$; 199(8-36)—at top of network, $x'$ should be $x$; 208(8-45)—at bottom of network, $w$ should be $w'$; 209(8-46)—in contact connected between $z'$ and $y$, $x$ should be $x'$. It is believed that all of the remaining four-variable circuits of [2] are correct.

For any switching function $f$, a parameter $m$, $0 \leq m \leq 16$, is defined as the number of configurations of the argument values for which $f=1$. Each function $f_1$ having $m_1<8$ is the complement of another function $f_2$ having $m_2=(16-m_1)$. If a planar contact network for $f_1$ is known, a network for $f_2$ can be found directly by means of a well-known construction [4], and conversely. Apparently, it is for this reason that Higgonet and Gréa gave networks only for functions having $m \leq 8$, although a number of their networks are nonplanar.

If the function $f_1$ has $m=8$, the complementary function $f_1'$ is in most cases equivalent to $f_1$ under transformations of the arguments. However, $f_1'$ is sometimes equivalent to another function $f_2$ having $m=8$. Of the 74 four-variable functions for which $m=8$, there are 16 pairs such that either member of each pair is equivalent to the complement of the other member. These 16 pairs of functions are listed in Table I (p. 198), and with each pair is given the transformation which must be applied to the first function in order to make it complementary to the second [1]. The facts summarized in Table I are apparently not generally realized. For example, the networks given by Higgonet and Gréa in [2] for the functions 175(8-12) and 181(8-18) use 8 and 6 contacts respectively. Since the networks are planar, it is evident that a 6-contact network for function 175 may be obtained from that for 181. The networks listed in [2] for the following other functions might similarly be simplified: 176(8-13), 192(8-29), 193(8-30), 206(8-43), 213(8-50), 214(8-51), 216(8-53), 218(8-55), 230(8-67), 232(8-69).

A second table of two-terminal contact networks, covering all of the 402 functions of four variables, has been compiled independently by Moore [5]. Moore's circuits are in many cases more economical than the correct circuits. However, application of the vector space synthesis method previously mentioned has produced nineteen networks which require in each case one

* Manuscript received by the PGEC, February 3, 1958. A portion of the work reported in this paper was done while the author held a Bell Telephone Labs. Fellowship at the Computation Lab., Harvard University, Cambridge, Mass.

† Deceased. Formerly at Comité d'Etude et d'Exploitation des Calculateurs Electroniques, Brussels, Belgium.

[1] In the original Harvard tabulation of the 402 functions, each was identified by an integer from the range 0, 1, · · · , 401. The identical list of functions was used by Higgonet and Gréa, but different identification numbers were assigned. In this paper, each function is identified by the Harvard number, followed by the Higgonet-Gréa number in parentheses. Functions having a Harvard number greater than 237 do not appear in the Higgonet-Gréa table, so for these only one number is given.

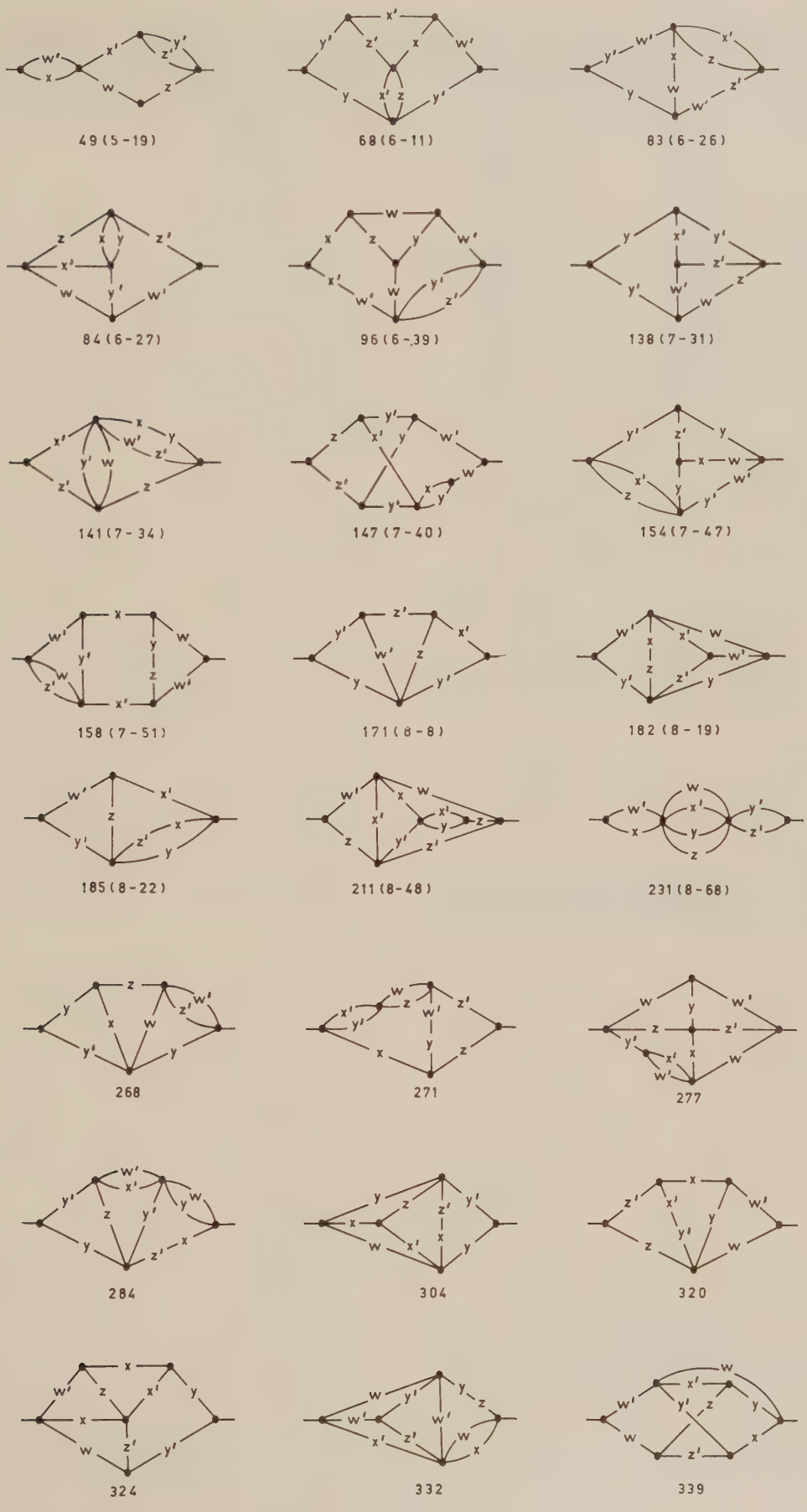



Fig. 1—Contact networks realizing various functions of four variables.

TABLE I

DISTINCT FOUR-VARIABLE FUNCTIONS WHICH ARE COMPLEMENTARY UNDER ARGUMENT TRANSFORMATIONS

| Functions | | Transformation |
|---|---|---|
| 172(8- 9) | 186(8-23) | $w'z'y'x$ |
| 173(8-10) | 217(8-54) | $z\,w'x'y'$ |
| 175(8-12) | 181(8-18) | $w'z'x'y'$ |
| 176(8-13) | 194(8-31) | $y'x'w'z'$ |
| 178(8-15) | 188(8-25) | $w'y\,x'z$ |
| 183(8-20) | 218(8-55) | $z\,w'x'y'$ |
| 187(8-24) | 207(8-44) | $x'w'y\,z'$ |
| 191(8-28) | 231(8-68) | $w'x\,y'z'$ |
| 192(8-29) | 211(8-48) | $w'y'z'x$ |
| 193(8-30) | 221(8-58) | $z\,y'w'x'$ |
| 196(8-33) | 213(8-50) | $w'y'z\,x$ |
| 197(8-34) | 230(8-67) | $x'z'y\,w$ |
| 199(8-36) | 232(8-69) | $w'x\,y\,z$ |
| 200(8-37) | 216(8-53) | $w'y\,z\,x$ |
| 206(8-43) | 223(8-60) | $y\,w'x'z'$ |
| 214(8-51) | 229(8-66) | $w'z\,x\,y$ |

fewer contact than the corresponding network of Moore and one or two fewer than that of Higgonet and Gréa [2]. These new networks, which are shown in Fig. 1, realize the following functions: 68(6-11), 84(6-27), 96(6-39), 138(7-31), 141(7-34), 147(7-40), 154(7-47), 158(7-51), 171(8-8), 182(8-19), 268, 271, 277, 284, 304, 320, 324, 332 and 339.[2] Moore [5] has established lower bounds for the number of contacts required to realize each of the 402 functions. It follows from these bounds that each of the circuits of Fig. 1 uses the minimum possible number of contacts.

Exhaustive application of the vector space synthesis method has indicated that a number of Moore's bounds may be raised. However, for nine of the 402 functions, the most economical networks now known require a number of contacts greater than the presently known lower bounds.

[2] The author has been informed that similar networks for eleven of these functions have been found previously: for 182 by S. H. Caldwell; for 96, 138, 141, 147, 154, 171, 182, 268, 271, 284, 332 by D. A. Huffman; for 96, 141, 171, 271, 332 by E. J. McCluskey, Jr.

TABLE II

LOWER BOUNDS* ON THE NUMBER OF CONTACTS REQUIRED TO REALIZE CERTAIN FOUR-VARIABLE FUNCTIONS

| Function | Lower Bound |
|---|---|
| 92(6-35) | 10 |
| 160(7-53) | 11 |
| 163(7-56) | 12 |
| 205(8-42) | 11 |
| 212(8-49) | 10 |
| 264 | 11 |
| 290 | 11 |
| 293 | 12 |
| 340 | 11 |

* Bounds due to E. F. Moore.

These nine functions with their present bounds are listed in Table II. Other circuit designers may be interested in attempting either to synthesize networks using the number of contacts shown in Table II, or to raise the bounds.

## BIBLIOGRAPHY

[1] H. H. Aiken and the Staff of the Computation Laboratory, "Synthesis of Electronic Computing and Control Circuits," Harvard University Press, Cambridge, Mass., pp. 231–278; 1951.
[2] R. Higgonet and R. Gréa, "Etude Logique des Circuits Electriques et des Systems Binaires," Editions Berger-Levrault, Paris, France, pp. 404–428; 1955.
[3] R. Gould, "The Application of Graph Theory to the Synthesis of Contact Networks," in "Proceedings of an International Symposium on the Theory of Switching 1957," to be published by Harvard University Press, Cambridge, Mass.
[4] Higgonet and Gréa, *op. cit.*, pp. 55–57.
[5] E. F. Moore, Bell Telephone Labs. Memorandum, 1952, unpublished.
[6] W. Keister, A. E. Ritchie and S. H. Washburn, "The Design of Switching Circuits," D. Van Nostrand Co., Inc., New York, N. Y., 1951.

# On the Loop- and Node-Analysis Approaches to the Simulation of Electrical Networks*

JOSEPH OTTERMAN†

*Summary*—The number of integrators in an analog-computer setup should be equal to the order of the differential equation describing the system. This paper presents a new procedure for tracing the loop currents which results in one-to-one correspondence between the number of integrators in the stimulation setup and the count of independent energy-storing elements in the network, *i.e.*, the degree of the system's characteristic equation. The generality of the procedure proves that it is always possible to trace the loop currents in such a way that excess integrators are avoided. The loop-analysis and the branch-variables-analysis approaches are discussed and examples given.

## Introduction

SIMULATION of an electrical network on an analog computer can be unsatisfactory because of hidden regenerative loops. Such regenerative loops may arise when the computer setup contains more integrators than the order of the differential equation describing the network. The regenerative loops are referred to as hidden loops, since the unwanted feedback arises because the actual computer components depart from exact, designated values. In other words, the computer solves a differential equation of a higher order than the physical system that is to be simulated; and the extraneous roots may cause considerable departures from the correct solutions. This problem has been pointed out and discussed by Walters[1] and more recently has been the subject of a communication by Scott.[2]

The danger of excess integrators has been avoided by Larrowe,[3] who developed the so-called "direct simulation" method. It has recently been pointed out[4] that excess integrators can be eliminated in the loop-analysis approach and in many cases in the node-analysis approach when the loops, or the reference node, are properly chosen. This paper presents a new procedure for tracing the loop currents in such a way that there is one-to-one correspondence between the number of integrators in the simulation setup and the count of independent energy-storing elements in the network, *i.e.*, the order of the differential equation describing the network.[5] The procedure is rather easy to apply, even to complicated networks or, through mechanical-electrical analogy, to mechanical systems. The generality of the procedure proves that it is always possible to trace the loop currents in such a way that excess integrators will be avoided. The node-analysis and the branch-variables-analysis approaches are discussed.

## Loop-Analysis Approach

For an example of a computer setup with an excess integrator, consider the simulation by the loop-analysis approach of the network discussed by Walters.[1] The network is shown in Fig. 1. If the loop currents are traced as in Fig. 1(a), the loop equations are:

$$\left(pL + R_1 + \frac{1}{pC_1}\right) I_1 - pL \cdot I_2 = E_i(p), \quad (1)$$

$$-pL \cdot I_1 + \left(pL + R_2 + \frac{1}{pC_2}\right) I_2 = 0 \quad (2)$$

where $p$ is the differentiation operator and where the initial conditions are disregarded. The computer setup, the possible instability of which has been analyzed by Walters, is shown in Fig. 2. This setup requires four integrators although there are only three energy-storing elements in the network.

When only one loop current is traced through the inductance $L$, as shown in Fig. 1(b), the loop equations are:

$$\left(R_1 + \frac{1}{pC_1}\right) I_1 + \left(R_1 + R_2 + \frac{1}{pC_1} + \frac{1}{pC_2}\right) I_2 = E_i(p) \quad (3)$$

$$-pL \cdot I_1 + \left(R_2 + \frac{1}{pC_2}\right) I_2 = 0. \quad (4)$$

The corresponding computer setup, shown in Fig. 3, requires only three integrators. The total number of amplifiers is reduced from 8 to 6.

The presentation which now will be given for tracing the loop currents to avoid excess integrators is compli-

* Manuscript received by the PGEC, February 3, 1958; revised manuscript received, June 30, 1958. This work was in part conducted by Project MICHIGAN under Dept. of the Army Contract (DA-36-039-SC-52654), administered by the U. S. Army Signal Corps.

† Willow Run Labs., University of Mich., Ypsilanti, Mich.

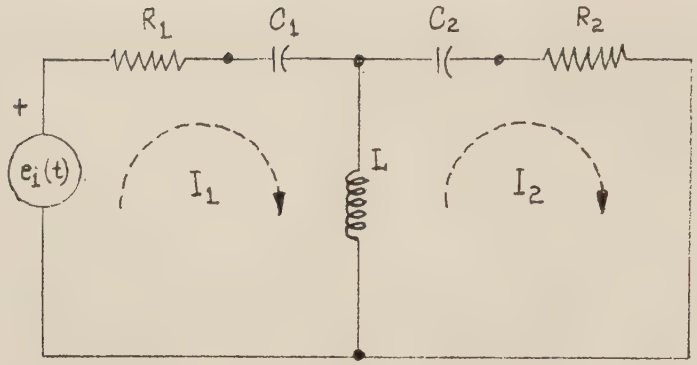
[1] L. G. Walters, "Hidden regenerative loops in electronic analog computers," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-2, pp. 1–4; June, 1953.

[2] N. R. Scott, "On the use of redundant integrators in analog computers," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 287–288; December, 1957.

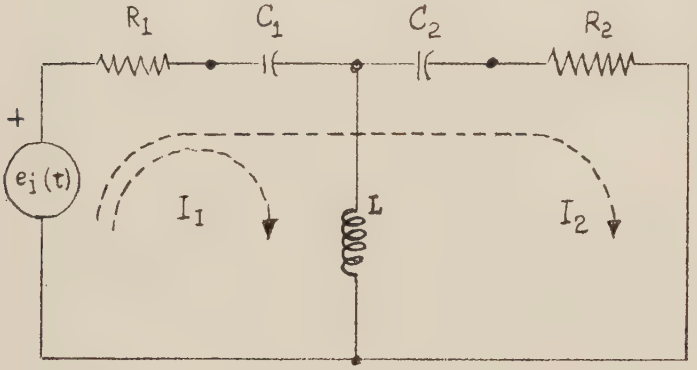
[3] V. L. Larrowe, "Direct simulation bypasses mathematics," *Control Engineering*, vol. 1, pp. 25–31; November, 1954.

[4] J. Otterman, "How to avoid extra integrators when simulating RLC networks," *Control Engineering*, vol. 4, pp. 111–114; November, 1957.

[5] J. Otterman, "On the order of the differential equation describing an electrical network," PROC. IRE, vol. 45, pp. 1024–1025; July, 1957.




(a) The loop equations contain terms with $I_1$ that range from $-1$ to $+1$ in powers of $p$; and terms with $I_2$ that range from $-1$ to $+1$ in powers of $p$.




(b) The loop equations contain terms with $I_1$ that range from $-1$ to $+1$ in powers of $p$; and terms with $I_2$ that range from $-1$ to 0 in powers of $p$.

Fig. 1—Different tracing of loop currents will result in eliminating an excess integrator in the computer setup.

cated by the fact that its aim is to demonstrate the generality of the procedure. The actual rules for tracing loop currents in such a way that excess integrators will be avoided [as was done in Fig. 1(b)] can be gleaned from this presentation by skipping all discussion pertaining to the count of nodes, elements, and separate parts.

In the following, a node is defined as a junction of two or more elements, a junction of two ends of the same elements, or an isolated end of a single element. All the current sources in the network are assumed to have been converted into voltage sources. The appearance of a voltage source in any loop does not affect the problem, and voltage sources are disregarded in the following discussion. Mutual inductances are not counted as elements and do not require any special consideration.

The procedure may best be described as an instruction to redraw the network according to certain directives. First draw all the resistances in the network. If any closed loops have been formed, trace loop currents through them. The number $M_r$ of the loops required will be given by the formula

$$M_r = B_r - N_r + S_r \qquad (5)$$

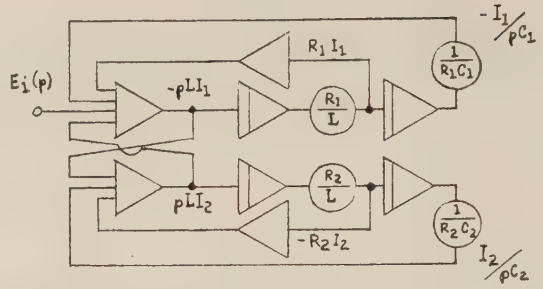



Fig. 2—Four integrators are required if loop currents are traced as in Fig. 1(a).

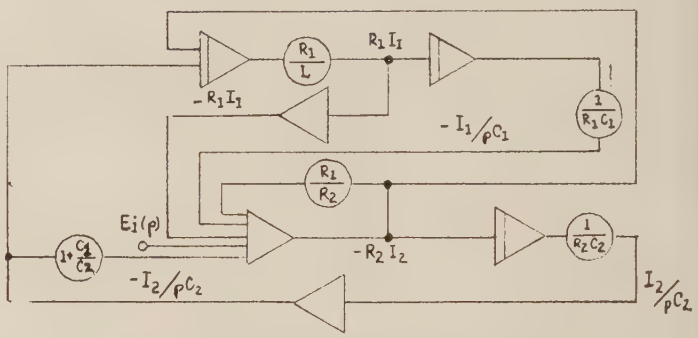



Fig. 3—Only three integrators are required if loop currents are traced as in Fig. 1(b).

where $B_r$, $N_r$ and $S_r$ are, respectively, the number of elements, the number of nodes, and the number of separate parts of the resistive network.[6] Since the loops contain resistances only, the loop equations are algebraic and no integrators are necessary to represent them.

Then, element by element, draw all the inductances and capacitances in an arbitrary order. During this process of increasing the partial drawing of the network, trace a loop current whenever a loop has been formed. An element or elements through which the loop current is traced for the first time, *i.e.*, an element or elements which are being added to the loop structure, will be referred to as "new" elements.

The formation of a new loop may occur in several topologically different ways, as is shown in Fig. 4. However, in each case the count of independent loops $M$ of the partially drawn network, defined by

$$M = B - N + S, \qquad (6)$$

where $B$ is the number of elements or branches, $N$ the number of nodes, and $S$ the number of separate parts in the partially drawn network, increases by only one. Thus, if we start with $M_r$ loops, given by (5), and trace one loop current each time a new loop is formed, we have always traced in the partial drawing of the net-

[6] The simplest counting procedure is to disregard all the elements already drawn which do not form part of closed loops. The count of $M_r$ by this procedure is identical to $M_r$ arrived at by counting the number of resistances, nodes, and separate parts in the resistive network, even though the counts of $B_r$, $N_r$ and $S_r$ are different.

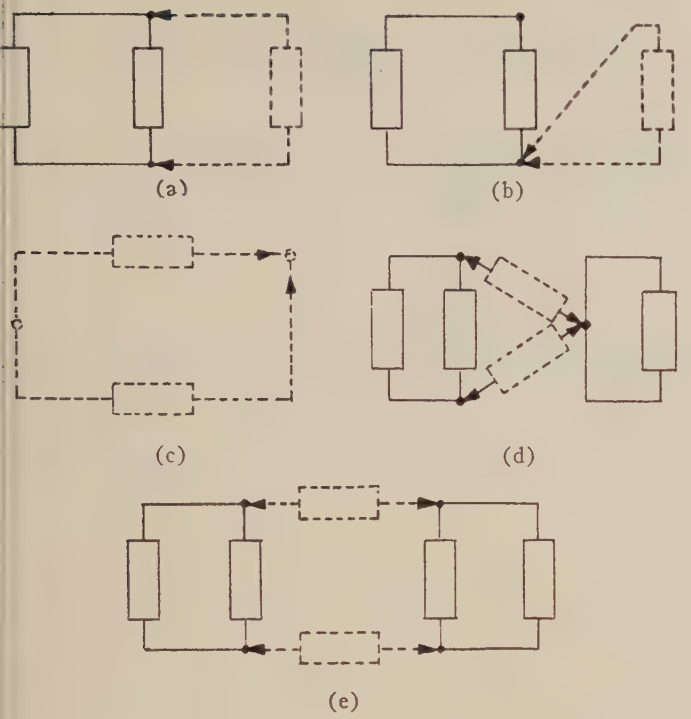



Fig. 4—The addition of new elements can occur in topologically different ways. The solid lines denote the previously drawn loop structure, the dotted lines denote the new elements. The rectangles denote an element or series of elements. In each case the count of the loops $M=B-N+S$ is increased by one.

work the number of loops given by (6). At the time we trace the loop current through the last new element (the addition of which completed the network), the right side of (6) becomes $B_c-N_c+S_c$, where $B_c$, $N_c$ and $S_c$ are, respectively, the number of elements or branches, the number of nodes, and the number of separate parts of the complete network; and we have traced the necessary number of $M_c$ loop currents. Moreover, the loop-current equations form an independent set. This is so because each loop current is passed through new elements; the equation for the voltages in this loop is therefore independent of equations written for the previously drawn loops structure.

The loop current is traced through new elements and [except in the case where new elements by themselves form a closed loop, as shown in Fig. 4(b) and Fig. 4(c)] through elements through which other loop currents have been traced. The completion of the current through the previously existing loop structure is done in such a way as to minimize the range in powers of $p$ of voltage terms associated with the new loop current. When the loop current passes through $L$ and $C$, or through $L$, $R$, and $C$, the expressions for voltages associated with the current include terms divided by $p$ and multiplied by $p$; and simulation requires two integrators. When the loop current passes through $L$ and $R$, or $R$ and $C$, the range in powers of $p$ of the associated terms for voltages is one; simulation requires one integrator. When the loop current passes through like elements, the terms for associated voltages exhibit the same power in $p$ ($p^{-1}$ when the loop consists of capacitance, $p^0$ when the loop consists of resistances, and $p^1$ when the loop consists of inductances), and simulation requires no integrators.

The procedure for minimizing the range in powers of $p$ for voltage terms associated with each new loop current treats separately seven cases, depending in part on what new elements are being added and in part on the possible choices of completing the loop through the previously traced loop structure. Examination of consecutive cases is required. It can be shown that in each case the additional number of integrators required is equal to the increase in the number of independent energy-storing elements.

*Case I. Addition of L Elements Only*

The new elements include inductances only. If it is possible to complete the loop through the previous loop structure in such a way that only inductances are traversed, then trace the loop structure through this path. The terms for voltages associated with this loop current exhibit the same power in $p$. No additional integrator is required in the computer analog of the network. This is in accordance with the fact that the added inductance forms part of an inductances loop and is therefore dependent in terms of Kirchhoff's law of voltages.[5] Its addition, therefore, does not increase the order of the differential equation of the drawn loop structure. If several inductances are being added, all but one are dependent in terms of Kirchhoff's law of currents as applied to the partially drawn network, so the increase in the order of the differential equation is still zero.[5]

If a closed loop through inductances only cannot be found, consider Case II and Case III.

*Case II. Addition of L or R and L Elements*

The new elements are inductances, or inductances and resistances. If a closed path through the previously drawn loop structure can be found that does not traverse any capacitances, then trace the loop current in such a way. Simulation of the current will require one integrator. This is in accordance with the fact that the addition of an independent inductance increases the order of the differential equation of the loop structure by one. If several inductances are being added, all but one are dependent in terms of Kirchhoff's law of currents as applied to the partially drawn network, so the increase in the differential equation is still one.

If a path that does not traverse any capacitance cannot be found, Case III applies.

*Case III. Addition of L or R and L Elements*

If a path that does not traverse any capacitance cannot be found, then trace a loop that includes a capacitance or capacitances. The new loop passes

through at least one inductance and one capacitance; thus the simulation requires two integrators. The capacitances in the previously drawn loop structure were not independent in terms of Kirchhoff's law of currents as applied to the partially drawn network, since a cut-set[7] of capacitances existed.[5] The addition of the new elements provides an additional current path, eliminating a cut-set of capacitances and converting one dependent capacitance to an independent capacitance. The increase of the order of the differential equation is two, since the counts of independent inductances and independent capacitances are increased each by one.

*Case IV. Addition of C Elements Only*

Read Case I, substituting the word "capacitance" for "inductance."

*Case V. Addition of C or R and C Elements*

Read Case II, substituting the word "capacitance" for "inductance," and vice-versa.

*Case VI. Addition of C or R and C Elements*

Read Case III, substituting the word "capacitance" for "inductance," and vice-versa.

*Case VII. Addition of L and C or L, R, and C Elements*

When both $L$ and $C$ are added, their addition cannot increase the number of loops of capacitances or inductances, as happens in Cases I and IV. The order of the differential equation is increased by two, since the counts of independent inductances and independent capacitances are increased each by one. Two integrators are required. The loop can be completed through the loop structure, which was previously traced, in an arbitrary manner.

[7] A cut-set is a set of elements such that when each element in the set is cut into two, the network is separated into two parts.

If the loop currents have been traced in accordance with the above procedure, the loops for stating Kirchhoff's law of voltages can be chosen in a different way than the loops of currents. In other words, adding and subtracting of loop equations does not affect the problem of minimizing the number of integrators.

Examples of the procedure for tracing the loop currents are given in Figs. 5 and 6 (pp. 203–205).

## Node-Analysis Approach

The node-analysis approach, too, quite often results in a higher number of integrators than the degree of the system's characteristic equation. Walters[1] analyzes the network shown in Fig. 1, with the node-voltage variables as shown in Fig. 7(a). The node equations are:

$$\left(\frac{1}{R_1}+pC_1\right)V_1 - pC_1 \cdot V_2 + 0 = \frac{E_i(p)}{R_1} \tag{7}$$

$$-pC_1 \cdot V_1 + \left(\frac{1}{pL}+pC_1+pC_2\right)V_2 - pC_2 \cdot V_3 = 0 \tag{8}$$

$$0 - pC_2 \cdot V_2 + \left(\frac{1}{R_2}+pC_2\right)V_3 = 0. \tag{9}$$

The corresponding computer setup, shown in Fig. 8, contains four integrators.

Excess integrators can be avoided by using the node-analysis approach if the network is an $LC$ continuous network,[4] that is, a line can be traced from a node to every capacitance in the network by traversing capacitances only, and a line can be traced from the same node to every inductance in the network by traversing inductances only. Such a node should be chosen as the reference node. In an $LC$ continuous network with only capacitances (only inductances) and resistances, any node adjoining a capacitance (an inductance) and a resistance can be chosen as the reference node.

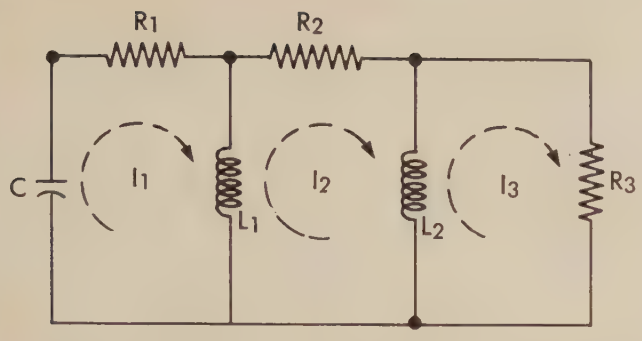
The network analyzed by Walters is actually an $LC$ continuous network. To avoid the danger of excess integrators, the node adjoining inductance $L$ and capacitances $C_1$ and $C_2$ should be chosen as the reference node. If the node-voltage variables are chosen as in Fig. 7(b), the node equations are:

$$\left(\frac{1}{R_1}+pC_1\right)V_1 - \frac{1}{R_1} \cdot V_2 + 0 = \frac{E_i(p)}{R_1} \tag{10}$$

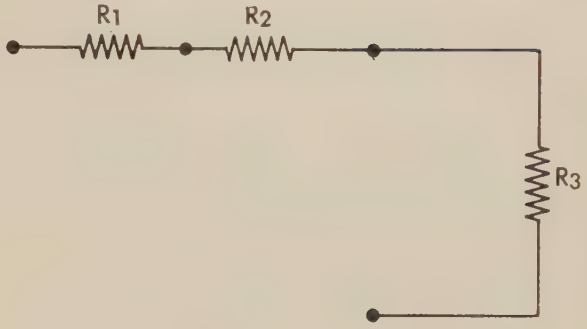
$$-\frac{1}{R_1} \cdot V_1 + \left(\frac{1}{pL}+\frac{1}{R_1}+\frac{1}{R_2}\right)V_2 - \frac{1}{R_2} \cdot V_3 = -\frac{E_i(p)}{R_1} \tag{11}$$

$$0 - \frac{1}{R_2} \cdot V_2 + \left(\frac{1}{R_2}+pC_2\right)V_3 = 0. \tag{12}$$

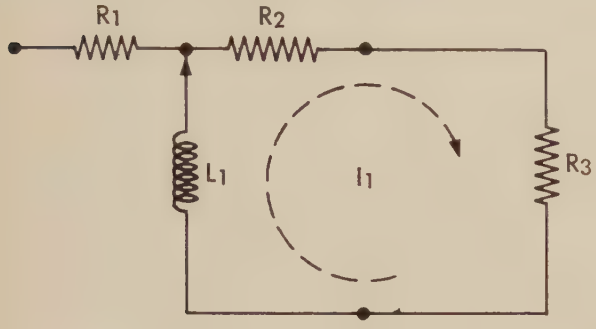
The computer setup, shown in Fig. 9, contains only three integrators.



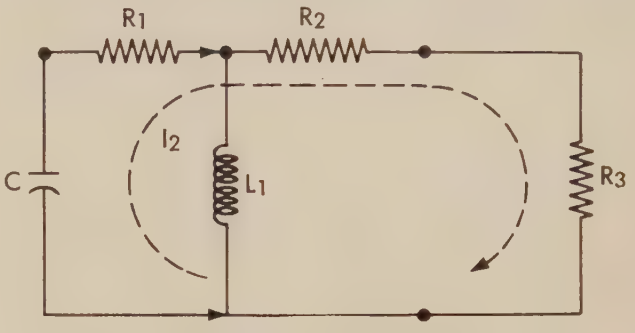

(a) Simulation of the loop currents requires four integrators if the loops are traced as above.



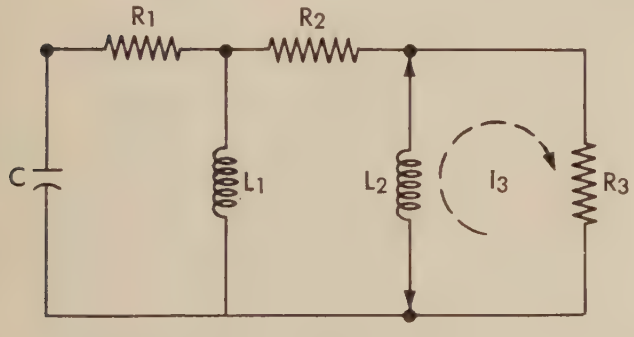

(b) The resistive network. No loops formed.



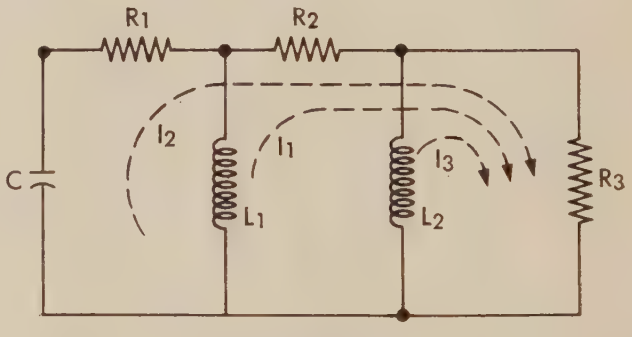

(c) Addition of $L_1$, $R_2$, $R_3$ to the loop structure. The new elements by themselves form a closed loop.




(d) Addition of $C$, $R_1$. Case V applies.




(e) Addition of $L_2$. Case II applies. The simpler of two possible loop currents was chosen.




(f) Simulation of the loop currents requires three integrators if the loops are traced as above.

Fig. 5—Tracing of loop currents through a simple network.

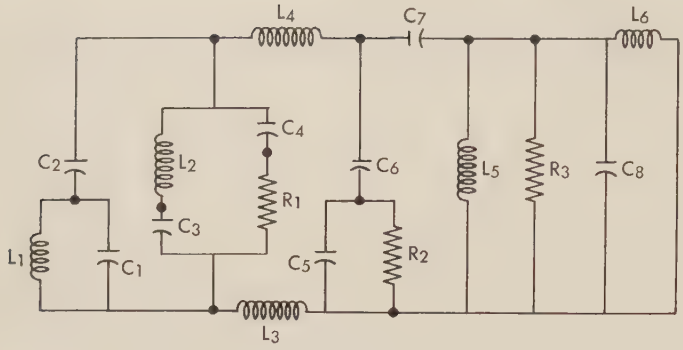
The network shown in Fig. 5 is an $LC$ continuous network, and in the node analysis the node common to $C$, $L_1$, $L_2$ and $R_3$ should be taken as the reference node, The network shown in Fig. 6 is not $LC$ continuous.

### Branch-Variables-Analysis Approach

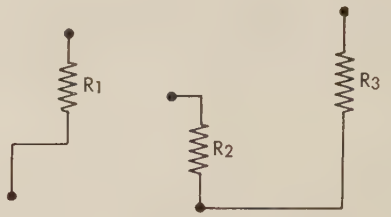
Both the node analysis and the loop analysis are systematical approaches based on the fundamental relations in electrical networks. Those fundamental relations consist of $2B$ equations, where $B$ is the number of branches or elements. The equations contain $2B$ unknowns, which are the current through each branch and the voltage across each branch. $B$ of the equations relate voltage across a branch to the current through the branch. $M$ equations relate voltages around $M$ independent loops. $N$-$S$ equations, where $N$ is the number of nodes and $S$ the number of separate parts, relate the currents through $N$-$S$ independent node-pairs, or, more generally, through $N$-$S$ independent cut-sets.

Analyzing the network in terms of those basic variables is more laborious, because the number of variables is larger; but it is, in some respects, much more flexible.

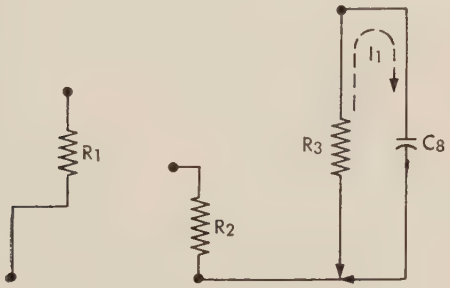
For an example of the branch-variables-analysis approach, consider the simulation of the network shown in Fig. 1. There are five elements in the network, and the number of variables is therefore 10. The positive direction of currents and voltage-drops is taken downwards and from left to right. The five equations relating voltages to currents are:



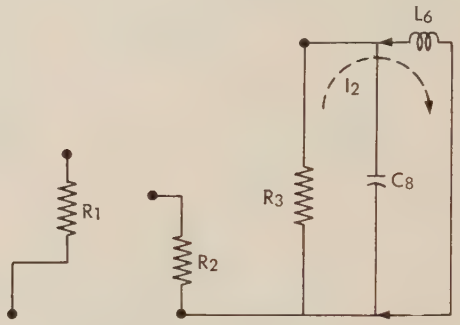

(a) The network to be simulated contains 14 energy storing elements. There is one cut-set of inductances, $L_3$, $L_4$, one cut-set of capacitances, $C_2$, $C_3$, $C_4$, $C_6$, $C_7$, one loop of inductances $L_5$, $L_6$ and one loop of capacitances $C_5$, $C_6$, $C_7$, $C_8$. Order of the differential equation is 10.



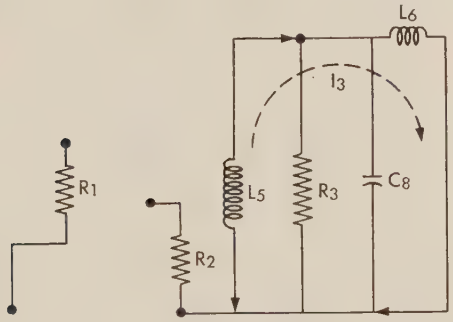

(b) The resistive network. No loops formed.



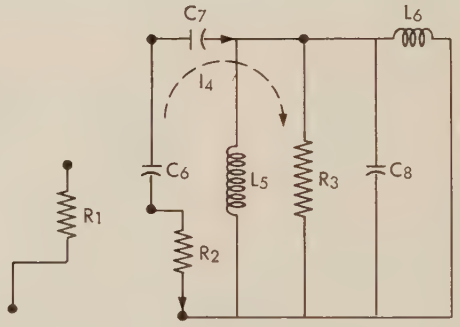

(c) Addition of $C_8$, $R_3$ to the loop structure. The new elements by themselves form a closed loop.



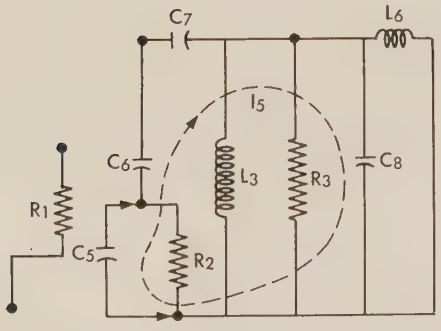

(d) Addition of $L_6$. Case II applies.



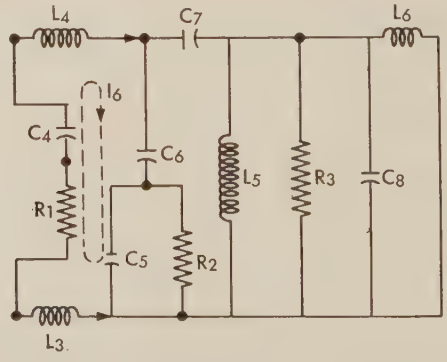

(e) Addition of $L_5$. Case I applies.




(f) Addition of $C_6$, $C_7$, $R_2$. Case V applies. The current could be traced through $C_8$ instead of $R_3$.

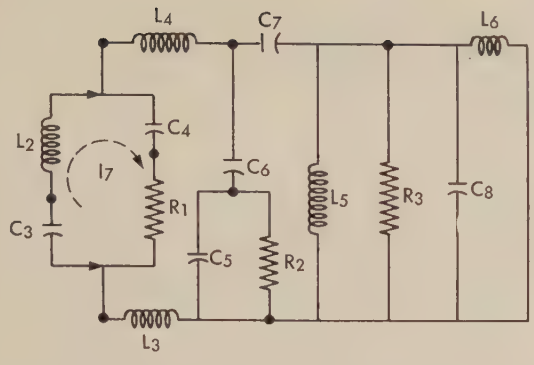



(g) Addition of $C_5$. Case IV applies.

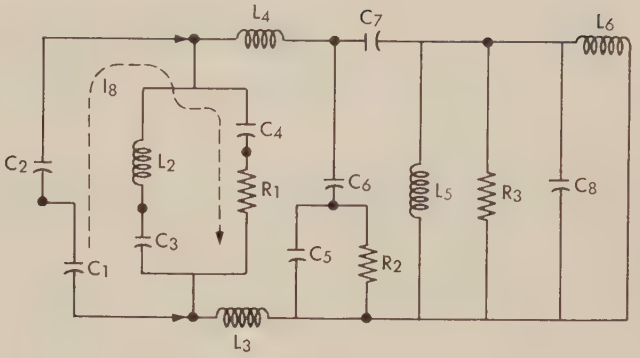



(h) Addition of $L_4$, $C_4$, $R_1$, $L_3$. Case VII applies. Loop current can be traced in an arbitrary way through the previously drawn loop structure.

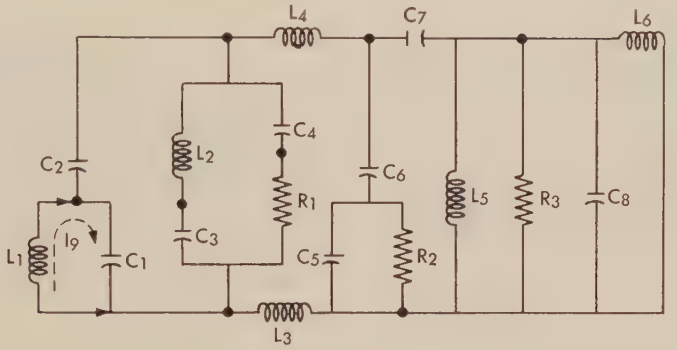
Fig. 6—Continued on next page.




(i) Addition of $L_2$, $C_3$. Case VII applies. Loop current can be traced in an arbitrary way through the previously drawn loop structure.

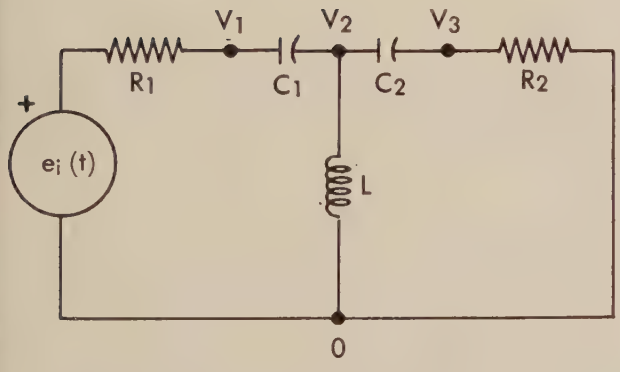



(j) Addition of $C_2$, $C_1$. Case V applies.

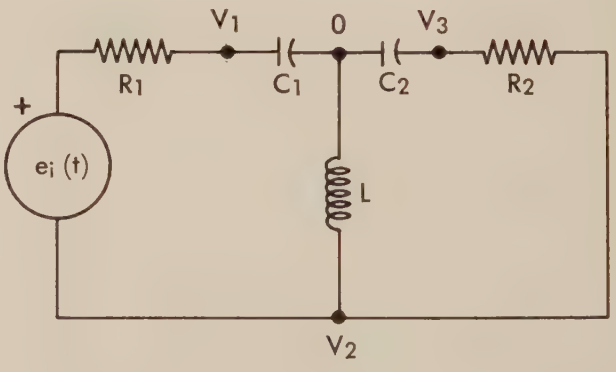



(k) Addition of $L_1$. Case III applies. Loop current can be traced in an arbitrary way through the previously drawn loop structure.

Fig. 6—Tracing the loop currents through a more complicated network.




(a) The node equations contain terms with $V_1$ that range from 0 to 1 in powers of $p$; terms with $V_2$ that range from $-1$ to 1 in powers of $p$; and terms with $V_3$ that range from 0 to 1 in powers of $p$.




(b) The node equations contain terms with $V_1$ that range from 0 to 1 in powers of $p$; terms with $V_2$ that range from $-1$ to 0 in powers of $p$; and terms with $V_3$ that range from 0 to 1 in powers of $p$.

Fig. 7—Different choice of reference node will result in eliminating an excess integrator in the computer setup.

$$I_{R1} = \frac{V_{R1}}{R_1} \tag{13}$$

$$V_{C1} = \frac{I_{C1}}{pC_1} \tag{14}$$

$$V_{C2} = \frac{I_{C2}}{pC_2} \tag{15}$$

$$I_{R2} = \frac{V_{R2}}{R_2} \tag{16}$$

$$I_L = \frac{V_L}{pL}. \tag{17}$$

The three equations relating currents through cut-sets are:

$$I_{R1} - I_{C1} = 0 \tag{18}$$

$$I_{C2} - I_{R2} = 0 \tag{19}$$

$$I_L + I_{C2} - I_{C1} = 0. \tag{20}$$

The two equations relating voltages in loops are:

$$E_i(p) + V_{R1} + V_L + V_{C1} = 0 \tag{21}$$

$$V_{C2} + V_{R2} - V_L = 0. \tag{22}$$

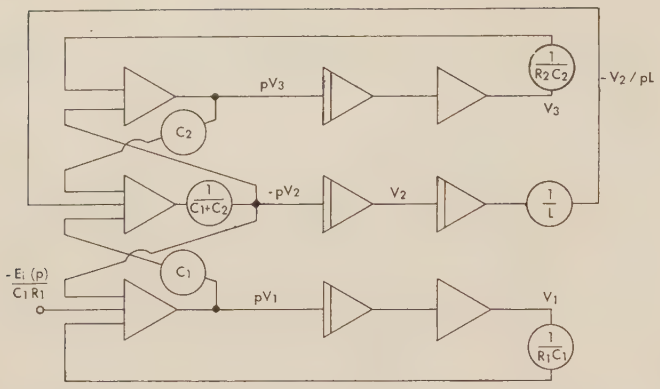



Fig. 8—Four integrators are required if the reference node is chosen as in Fig. 7(a).

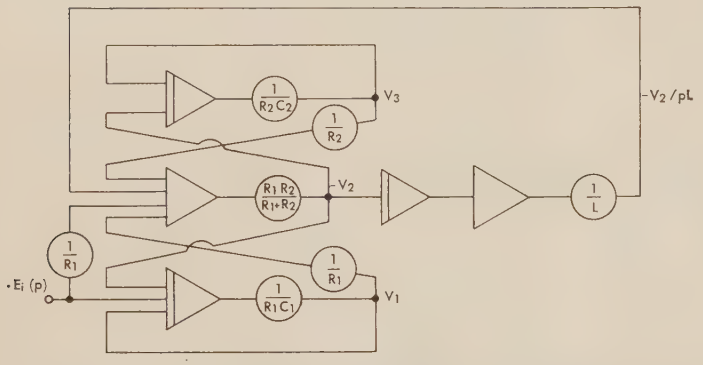



Fig. 9—Only three integrators are required if the reference node is chosen as in Fig. 7(b).

The corresponding computer setup is shown in Fig. 10. This setup could also be obtained by applying the technique of Larrowe.[3]

To avoid extra integrators when a cut-set of inductances or capacitances occurs, the equation expressing the voltage across one element in the set in terms of the current through this element should be replaced by an equation expressing the same voltage in terms of voltages across other elements in the set. For instance, in the cut-set of $L_3$ and $L_4$ in the network shown in Fig. 6(a), the element $L_3$ is arbitrarily chosen as a dependent element and the equation for the voltage across this element

$$V_{L3} = pL_3I_{L3} \tag{23}$$

is replaced by

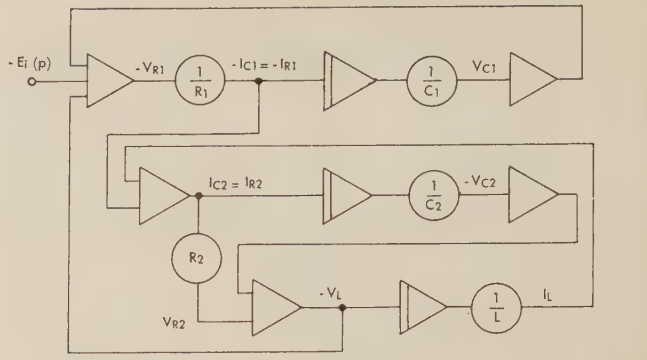



Fig. 10—The network of Fig. 1 can be simulated by this setup in which all the branch-variables are available for recording purposes.

$$V_{L3} = -pL_3I_{L4} = -\frac{L_3}{L_4}V_{L4}. \tag{24}$$

Conversely, when a loop of inductances or capacitances occurs, the equation expressing the current through one element in the set should be replaced by an equation expressing the same current in terms of currents through other elements in the set. For instance [Fig. 6(a)] the equation for the current

$$I_{L6} = \frac{V_{L6}}{pL_6} \tag{25}$$

is replaced by

$$I_{L6} = \frac{V_{L5}}{pL_6} = \frac{L_5I_{L5}}{L_6}. \tag{26}$$

Using this technique, no integrators are associated with the energy-storing elements chosen as the dependent elements.

## Conclusion

The loop-analysis, node-analysis, and branch-variables-analysis approaches have been reviewed. The choice of the most suitable approach depends on many factors, such as the relative number of loops and nodes, the presence or absence of mutual inductances, and the desirability of representing voltages across elements or voltages relative to a certain point in the network. Therefore, familiarity with the different approaches is distinctly advantageous.

# Generalized Parity Checking*

HARVEY L. GARNER†

*Summary*—**The usual definition given for the parity check is unwieldy and not particularly suited for the analysis or the study of the arithmetic properties of parity checking. The definition of parity by means of congruences provides a convenient mathematical basis for the concepts of the parity check. In this paper congruence notation is used to generalize the concepts of parity to include nonbase two number systems. Consideration is given to the cases where the check base is equal to the number base, and where it is not equal to the number base. The arithmetic properties of each case are considered by means of congruences.**

## Introduction

THE CONCEPT of the parity check[1] is widely known and has frequently been used to detect and in some cases to correct errors in digital systems. The parity check is obtained by associating with each representation of information an auxiliary digit or group of digits known as the parity check digit(s). The proper choice of the relationship between the check digits and the information digits permits the establishment of mathematical operations, which may indicate erroneous digits or provide correction data. Parity checking usually is defined for the binary system in terms of the odd or evenness of the number of ONE digits in a specified block of binary digits. If the number of ONE digits is even the parity check digit is a zero. If the number of ONE digits is odd the parity check digit is a one. This definition is sufficient for most practical applications of the parity concept, but gives no insight into the fundamental principles of parity. A restatement of parity in terms of linear congruences provides a mathematical basis for the parity concept. In terms of congruences, the parity check may be defined for the general case where the check base does not equal the number base. The parity check which employs a check base equal to the number base (as in the usual definition given above) is a special case of the general parity check. The congruence notation also facilitates the study of the arithmetic properties of the parity check. The properties of the parity check with regard to arithmetic operations has received little attention in the literature of error correcting and detecting codes. Previous research in this area is due to Block *et al.*[2,3] and Robertson.[4] Robertson studied the application of a Hamming error correcting code in a general-purpose digital computer. The papers by Block describe a particular error detecting scheme employed in the RAYDAC computer.

The primary contribution of this paper is the unification of the parity check concepts within the framework of congruences.

## Congruences

The remainder of this paper makes considerable use of congruence notation. A short presentation of pertinent concepts and relationships is included at this point. Additional material on the subject may be found in any standard text on number theory.[5]

The congruence

$$A \equiv \alpha \text{ Mod } b \tag{1}$$

is read "$A$ is congruent to $\alpha$ modulo $b$." The congruence relationship states that

$$A = \alpha + bt \tag{2}$$

is valid for some value of $t$, where $A$, $\alpha$, $b$ and $t$ are integers. $\alpha$ is called the residue and $b$ the base or modulus of the number $A$.

As examples of congruences, consider

$$10 \equiv 7 \text{ Mod } 3 \tag{3}$$

$$10 \equiv 4 \text{ Mod } 3 \tag{4}$$

$$10 \equiv 1 \text{ Mod } 3. \tag{5}$$

In these examples the integers 7, 4, and 1 form a residue class of 10 Mod 3. Of particular importance is the least positive residue of the class which in this example is one. The least positive residue is that residue for which $0 \leq \alpha \leq b$.[6]

Unless otherwise specified, the term residue as used in this paper means the least positive residue.

For the most part, congruences may be treated in the same manner as the equality sign in ordinary arithmetic. The main exception pertains to division. The properties of congruences used here are presented below without proof.

1) Congruences to the same modulus may be added and the result is a valid congruence.

$$\sum_{i=1}^{n} A_i \equiv \left( \sum_{i=1}^{n} \alpha_i \right) \text{Mod } b. \tag{6}$$

* Manuscript received by the PGEC, March 10, 1958; revised manuscript received, June 23, 1958.

† University of Michigan, Ann Arbor, Mich.

[1] R. W. Hamming, "Error detecting and correcting codes," *Bell Sys. Tech. J.*, vol. 26, pp. 147–160; April, 1950.

[2] R. M. Block, R. V. D. Cambell, and M. Ellis, "The logical design of the raytheon computer," *Math. Tables and Aids to Comp.*, vol. 3, pp. 286–296, 317–323; October, 1948.

[3] R. Block, U. S. Patent No. 2,634,052.

[4] J. E. Robertson, "Error Detection and Correction in Binary Parallel Digital Computers," Digital Computer Lab., University of Illinois, Urbana, Ill., "Electronic Digital Computer" Internal Rep. No. 37, pp. 70–81; 1952.

[5] G. H. Hardy, and E. M. Wright, "An Introduction to the Theory of Numbers," Oxford University Press, London, Eng.; 1956.

[6] The equality sign may exist on only one side of the inequality.

2) Congruences to the same modulus may be multiplied and the result is a valid congruence.

$$\prod_{i=1}^{n} A_i \equiv \left( \prod_{i=1}^{n} \alpha_i \right) \text{Mod } b. \tag{7}$$

3) Congruences are transitive. If $A \equiv B$ and $B \equiv C$ then $A \equiv C$.

4) The following congruence relationships provide a basis for numerical checking procedures

$$\sigma b^n \equiv \sigma \text{ Mod } (b-1) \tag{8}$$

$$\sigma b^n \equiv +\sigma \text{ Mod } (b+1) \quad n \text{ even} \tag{9}$$

$$\equiv -\sigma \text{ Mod } (b+1) \quad n \text{ odd.} \tag{10}$$

## Generalization of Parity

A parity check need not necessarily include every digit of the number. In the general case, multiple parity checks may be employed, each check concerning only specific digits. However, for purposes of simplification, only the case of a single parity check over all digits is considered here. The parity of a number from a consistently weighted number system is defined now. Consider the number

$$\sigma_n \sigma_{n-1} \cdots \sigma_2 \sigma_1 \tag{11}$$

This is the usual shorthand notation for the polynomial representation of a number $N$ in a consistently weighted number system, base $b$:

$$N = \sigma_n b^{n-1} + \sigma_{n-1} b^{n-2} + \cdots + \sigma_2 b^1 + \sigma_1 b^0. \tag{12}$$

A general parity check digit $p$ associated with number $N$ is defined by

$$F(N) \equiv p \text{ Mod } m \tag{13}$$

where $p$ is the least positive residue.

This paper considers two different functions of $N$. The first function is

$$F(N) = N \tag{14}$$

and the parity check digit $p$ is given by

$$N \equiv p \text{ Mod } m \qquad m \neq b. \tag{15}$$

This type of check is based directly on the numerical properties of linear congruences and the check digit is some function of the magnitude of the number $N$. This type of check is called a numerical parity check.

If the check base $m$ is identical to the number base $b$, then

$$N \equiv \sigma_1 \text{ Mod } b. \tag{16}$$

The check is unsatisfactory because it is a function of only the low order digit of the number representation. This is due to the fact that

$$\sigma_i b^n \equiv 0 \text{ Mod } b \qquad n > 0. \tag{17}$$

A check for the case $m = b$ is obtained if

$$F(N) = \sigma_n + \sigma_{n-1} + \cdots + \sigma_2 + \sigma_1. \tag{18}$$

The parity check digit $p$ for this type of check, which is termed digital checking, is defined by the relation

$$\sum_{i=1}^{n} \sigma_i \equiv p \text{ Mod } b. \tag{19}$$

An equivalent definition may be given in terms of ring addition. Ring addition modulo $b$ is specified by the symbol $\oplus$. Ring addition is simply normal addition without carry, for example,

$$(b-1) \oplus 1 = 0. \tag{20}$$

In terms of ring addition modulo $b$ the parity digit $p$ is given as

$$p = \sigma_n \oplus \sigma_{n-1} \oplus \cdots \oplus \sigma_2 \oplus \sigma_1. \tag{21}$$

If the number system is binary, and the check base is two, then the previous definition for the digital check provides a parity digit which makes the total number of ONE digits of the number plus check representation even in correspondence with the usual parity procedures. The definition is extended easily to an odd parity check. For this case the parity check digit is given by $\bar{p}$, the diminished radix complement of $p$ defined as

$$\bar{p} \oplus p = b - 1. \tag{22}$$

As an example of the parity definition for $m = b$, consider the following.

Given 12894, a number in base 10, then

$$(1 + 2 + 8 + 9 + 4) \equiv p \text{ Mod } 10 \tag{23}$$

or

$$p = 1 \oplus 2 \oplus 8 \oplus 9 \oplus 4 = 4. \tag{24}$$

Both types of parity checks are restricted in application to the detection of certain classes of errors in the number representation. Error correction is only possible if multiple parity checks are employed.[7,8] The digital parity check, $m = b$, detects all alterations except the alterations in which the sum of the digit perturbations is congruent to 0 modulo $b$. Let the decimal number of the previous example be corrupted by $+5$ in the first column and by $-5$ in the third column. The result is an erroneous representation but the error is obviously undetectable.

$$1 \oplus 2 \oplus 3 \oplus 9 \oplus 9 = p = 4. \tag{25}$$

A numerical parity check Mod $m$, $m \neq b$, is insensitive to an error if the magnitude of the error is congruent to zero Mod $m$. Thus the error is detectable only if the check base $m$ is not a divisor of the magnitude of the error. The magnitude of the error in the previous example is 495. The error is not detected by Mod 3, 5, 9 or 11 checks.

Parity checks are particularly effective in detecting

[7] M. J. E. Golay, "Notes on digital coding," Proc. IRE, vol. 37, p. 657; June, 1949.

[8] M. J. E. Golay, "Binary coding," IRE Trans. on Information Theory, No. PGIT-4, pp. 23–28; September, 1954.

errors for which only one digit has been modified. For this class of errors a parity check $m=b$ is always effective. If $m$ is less than $b$ then perturbations congruent to zero modulo $m$ are undetectable.

## Arithmetic Invariance of the Digital Parity Check

In the arithmetic system employed in most digital machines, a one-to-one correspondence is established between the machine digit representation and a set of positive and negative fractions. The rules of machine arithmetic operations are set up in such a way as to preserve this one-to-one correspondence under the operation of addition. Addition is the fundamental arithmetic operation of the machine and the other arithmetic operations, subtraction, multiplication and division, are defined in terms of addition and the auxiliary operations, shift and complement.

It is obvious that the digital parity check is invariant to the shift operation if no digits are deleted or added to the representation. This is true because the digital parity check is a function of the modulo sum of the individual digits rather than a function of the magnitude of the representation. Also, the parity check is independent of the position of the radix point. For this reason the remaining discussion does not explicitly consider whether the representations are in correspondence with fractions or integers.

Consider now the addition of two consistently weighted numbers in base $b$ representation. The check base, of course, is also equal to $b$.

$$A + D = S$$

or

$$\begin{array}{r} a_n \cdots a_1 \\ +\ d_n \cdots d_1 \\ \hline c_{n+1} s_n \cdots s_1 \end{array}.$$

Now

$$s_i = a_i \oplus d_i \oplus c_i \tag{26}$$

and

$$(a_i + d_i + c_i) = s_i + c_{i+1} b \tag{27}$$

$$c_1 = 0 \tag{28}$$

$$c_i = 0 \text{ or } c_i = 1 \text{ if } i \neq 1; \tag{29}$$

$c_i$ is the $i$th carry digit from the $i$th minus one column.

The parity check digit $p$ for the sum is given as

$$c_{n+1} \oplus s_n \oplus s_{n-1} \oplus \cdots \oplus s_1 = p_S \tag{30}$$

or

$$c_{n+1} \oplus a_n \oplus d_n \oplus c_n \oplus \cdots \oplus a_1 \oplus d_1 = p_S, \tag{31}$$

but

$$a_n \oplus \cdots \oplus a_1 = p_A \tag{32}$$

and

$$d_n \oplus \cdots \oplus d_1 = p_D; \tag{33}$$

therefore,

$$c_{n+1} \oplus \cdots c_2 \oplus p_A \oplus p_D = p_S. \tag{34}$$

The last expression provides a means of checking the addition operation in terms of the parity check digits of the sum, the addend, and the augend and the generated carries. Alternatively, the parity check may be in terms of $p'$, the radix complement or additive inverse of $p$ defined as

$$p' \oplus p = 0. \tag{35}$$

If $p'$ rather than $p$ is used as the check digit, the check procedure for addition is given as

$$c_n \oplus \cdots \oplus c_1 \oplus p_S' = p_A' \oplus p_D'. \tag{36}$$

The class of addition errors detected by the digital parity check is the same as the class of representation errors defined in the previous section. However, it must be remembered that the logic of the usual addition circuits is such that the various carry digits are not independent. The parity checking procedure is valid only if the carry digits are correct.

Consider the decimal addition of the following numbers:

| | | | | | $p$ |
|---|---|---|---|---|---|
| 6 | 6 | 9 | 4 | 1 | 6 |
| 3 | 2 | 4 | 5 | 1 | 5 |
| 9 | 9 | 3 | 9 | 2 | 2 |

Due to the occurrence of one carry, the check of addition is

$$6 \oplus 5 \oplus 1 = 2$$
$$2 = 2. \tag{37}$$

The check of the complement operation is straightforward. The diminished radix complement of a representation $A$ is obtained by substituting for each digit $a_i$ the diminished radix complement, $\bar{a}_i$.

$$a_i \oplus \bar{a}_i = b - 1 \tag{38}$$

given

$$A = a_n \cdots a_1 \quad \text{and} \quad \overline{A} = \bar{a}_n \cdots \bar{a}_1 \tag{39}$$

$$p_A = a_n \oplus \cdots \oplus a_1 \quad \text{and} \quad p_{\overline{A}} = \bar{a}_n \oplus \cdots \oplus \bar{a}_1 \tag{40}$$

and

$$a_n \oplus \bar{a}_n \oplus \cdots \oplus a_1 \oplus \bar{a}_1 = p_A \oplus p_{\overline{A}} = f. \tag{41}$$

Then

$$f \equiv n(b-1) \bmod b \tag{42}$$

$$(f + n) \equiv nb \bmod b \tag{43}$$

$$(f + n) \equiv 0 \bmod b \tag{44}$$

and

$$p_{\bar{A}} + p_{\bar{A}} + n \equiv 0 \text{ Mod } b, \tag{45}$$

or

$$p_{\bar{A}} \oplus p_A \oplus n \text{ Mod } b = 0. \tag{46}$$

A check of the diminished radix complement consists of the modulo $b$ sum of the parity check of the normal representation, the parity check of the complemented representation and the number of digits comprising the representation. In the special case where $b=2$, the check of the complemented representation is equal to the check of the normal representation if $n$ is even, and opposite if $n$ is odd.

A check of the radix complement operation is somewhat more complex than that required for the diminished radix complement. The complications are due to the definition of the radix complement rather than to the nature of the parity check procedure. Two checking procedures are possible. The first is based on the fact that the radix complement $A'$ is obtained from the diminished radix complement $\overline{A}$ by the addition of one to the lowest order digit. This method requires cognizance of the carries produced when the one is added to the lowest order digit. The check is given by the relationship

$$p_{A'} = 1 \oplus p_{\bar{A}} \oplus \sum_{i=2}^{n+1} c_i \text{ Mod } b. \tag{47}$$

The second procedure for obtaining the radix complement check is based upon the well-known rule for obtaining the digitwise radix complement: "Starting with the low order digit, radix complement the first non-zero digit; the remaining higher order digits are changed by diminished radix complementation." This method requires a knowledge of the number of zeros preceding the first nonzero digit. Let this number be $q$.

Given

$$A = a_n \cdots a_{q+2} a_{q+1} a_q \cdots a_1 \tag{48}$$

$$A' = \bar{a}_n \cdots \bar{a}_{q+2} a'_{q+1} a_q \cdots a_1 \tag{49}$$

$$a_i = 0 \text{ if } i \leqq q \tag{50}$$

so

$$p_A = a_n \oplus \cdots \oplus a_{q+2} \oplus a_{q+1} \tag{51}$$

$$p_{A'} = \bar{a}_n \oplus \cdots \oplus \bar{a}_{q+2} \oplus a'_{q+1} \tag{52}$$

$$p_A \oplus p_{A'} = a_n \oplus \bar{a}_n \oplus \cdots \oplus a_{q+2} \oplus \bar{a}_{q+2} \oplus a_{q+1} \oplus a'_{q+1}, \tag{53}$$

$$p_A \oplus p_{A'} = [(n-q-1)(b-1)+b] \text{ Mod } b \tag{54}$$

or

$$p_A \oplus p_{A'} \oplus (n-q) \text{ Mod } b = 1. \tag{55}$$

Multiplication consists of repetitive addition with appropriate shift operations. The result is a double-length product. The digital parity of the double-length product is found from consideration of the addition operations since parity is not affected by the shift operation. The linear congruence is multiplicative. The check procedure for the product $P=A\times D$ is

$$p_p = (p_A \odot p_D) \oplus \sum c_i \text{ Mod } b \tag{56}$$

or

$$(p_A \times p_D) + \sum c_i \equiv p_p \text{ Mod } b. \tag{57}$$

The symbol $\odot$ indicates multiplication, modulo $b$. It is observed that the digital parity check provides no protection against faulty shift operations. Parity protection against this type of error is obtained if multiple checks are employed.

The division algorithm commonly employed is easily explained in terms of the remainder theorem.

Consider

$$\frac{X}{Y} = Q + \frac{R}{Y} \tag{58}$$

or

$$X = QY + R \tag{59}$$

$$R = X - QY \tag{60}$$

where

$$Q = q_n b^{n-1} + \cdots + q_j b^{j-1}. \tag{61}$$

Then

$$R = X - (q_n b^{n-1} + \cdots + q_j b^{j-1}) Y. \tag{62}$$

The subtraction of $Y$ from $X$, $q_i$ times is accomplished by the addition of $\overline{Y}$, $q_i$ times. Carries which are generated must be considered. The parity check expression is given as

$$p_R = p_X \oplus (p_Q \odot p_{\overline{Y}}) \oplus \sum c_i \text{ Mod } b. \tag{63}$$

This check may be employed for each division operation or only as a check of the over-all division process. In any case the check is insensitive to shift errors.

*Examples of Arithmetic Checking* ($m=b=10$)

Addition

| | | | | |
|---|---|---|---|---|
| | 3 | 4 | 6 | ③ |
| + | 4 | 5 | 8 | ⑦ |
| | 8 | 0 | 4 | ② |

○ the circled digits are the check digits

Check

$$p_A \oplus p_D \oplus \sum c_i \text{ Mod } b = p_s. \tag{64}$$

$$3 \oplus 7 \oplus 2 = 2 \tag{65}$$

Complement

nines complement

$A = 0345②$

$A = 9654④$

Check

$$p_{\bar{A}} \oplus p_A \oplus n \bmod b = 0 \quad (66)$$

$$4 \oplus 2 \oplus 4 = 0 \quad (67)$$

ten's complement

$A = 034500$ ②

$A' = 965500$ ⑤

Check

$$p_{A'} \oplus p_A \oplus (n - q) \bmod b = 1 \quad (68)$$

$$5 \oplus 2 \oplus 4 = 1 \quad (69)$$

Product

| | |
|---|---|
| 346 ③ | |
| 213 ⑥ | |
| 346 | |
| 346 | |
| 692 | |
| 346 | |
| 1038 | 5 carries |
| 346 | generated |
| 4498 | |
| 346 | |
| 39098 | |
| 346 | |
| 73698 ③ | |

Check

$$p_P = (p_A \odot p_D) \oplus \sum c_i \bmod b \quad (70)$$

$$3 = (3 \odot 6) \oplus 5 \quad (71)$$

$$3 = 8 \oplus 5 = 3$$

Division

$$Q = \frac{073699\ ④}{0346\ ③} = \frac{X}{Y}$$

the ten's complement of 0346 ③ is 9654 ④

| | Carries | | Carries |
|---|---|---|---|
| 073699 | 2 | | |
| 9654 | | | |
| 039099 | 2 | 0347 | 3 |
| 9654 | | 9654 | |
| 004499 | | 0001 | |
| 9654 | | 9654 | |
| 969899 | | 9655 | |
| 04499 | 3 | 001 | |
| 9654 | | | |
| 01039 | | $Q = 213$ ⑥ | |
| 9654 | | $R = 1$ | |
| 97579 | | $\sum c_i = 13$ | |
| 1039 | 1 | | |
| 9654 | | | |
| 0693 | 2 | | |
| 9654 | | | |

Check

$$p_R = p_X \oplus (p_Q \odot p_{\bar{Y}}) \oplus \sum c_i \bmod b$$

$$1 = 4 \oplus (6 \odot 4) \oplus 3 = 4 \oplus 4 \oplus 3 = 1. \quad (72)$$

## Numerical Checking, Mod $m(m \neq b)$, of Arithmetic Operations

The modulus of the numerical parity check is not the same magnitude as the radix of the number representation. Congruence relationships are valid for any modulus. However, only certain moduli yield procedures which permit straightforward computation of the parity digits. There exist in particular two interesting and useful checks based on procedures congruent modulo $(b-1)$ and congruent modulo $(b+1)$. These checks are called the diminished base numerical check and the augmented base numerical check, respectively. The simplest checks of the numerical class depend directly on certain congruence relationships. Numerical checking procedures have one very desirable feature not associated with digital checking procedures; they do not require a tally or cognizance of the carries generated by the addition process.

### The Diminished Base Numerical Check

The diminished base numerical check is dependent on the following properties of linear congruences:

$$\sigma_i b^n \equiv \sigma_i \bmod (b - 1). \quad (73)$$

This relationship permits parity calculation independent of the digit position or weight. Consider the sum

$$X + Y = S$$

$$X = x_n b^{n-1} + \cdots + x_1 b^0 \quad (74)$$

$$Y = y_n b^{n-1} + \cdots + y_1 b^0 \quad (75)$$

$$S = c_{n+1} b^n + s_n b^{n-1} + \cdots + s_1 b^0. \quad (76)$$

Since congruences are additive,

$$X = \sum_{i=1}^{n} x_i b^{i-1} \equiv \sum_{i=1}^{n} x_i \bmod (b - 1). \quad (77)$$

The parity check digit $p_X$ associated with the representation $X$ is defined by the term

$$\sum_{i=1}^{n} x_i$$

reduced to the least positive residue of the class. The parity check digit $p_X$ also may be defined by

$$p_X = x_n \oplus \cdots \oplus x_1 \quad (78)$$

for either definition

$$X = \sum_{i=1}^{n} x_i b^{i-1} \equiv p_X \bmod (b - 1). \quad (79)$$

The check digit $p_Y$ is given as

$$Y = \sum_{i=1}^{n} y_i b^{i-1} \equiv p_X \bmod (b - 1). \quad (80)$$

Linear congruences are additive, therefore,

$$X + Y = \sum_{i=1}^{n} (x_i + y_i)b^{i-1} \equiv (p_X + p_Y) \text{ Mod } (b-1) \quad (81)$$

and

$$p_S = p_X \oplus p_Y. \quad (82)$$

The modulo $(b-1)$ check is numerical and hence does not require cognizance of the generated carries. Nevertheless, the check is sensitive to errors in carry generation and propagation. The check detects all digit alterations for which the modulo $(b-1)$ sum of the changes does not equal zero. This is equivalent to the statement that an alteration is detectable only if $b-1$ is not a factor of the magnitude of the arithmetic error. A single carry failure causes an alteration of minus one and is therefore detectable. However, the error is not correctable on the basis of the parity information.

A check of the complement operation is obtained from the basic definition of the complemented representation. The definitions which follow are for representations of $n+m+1$ digits. The representation has $n$ digits to the left of the radix point, $m$ digits to the right of the radix point and one sign digit. The diminished radix complement of $X$ is $\overline{X}$ where

$$X + \overline{X} = b^{n+1} - b^{-m}. \quad (83)$$

The radix complement $X'$ is given as

$$X + X' = b^{n+1}. \quad (84)$$

The usual congruence relationships are defined for integer values. It is, therefore, convenient to consider the radix point on the extreme right of the representation. The check is obtained by observing that

$$b^m(b^{n+1} - b^{-m}) \equiv 0 \text{ Mod } (b-1) \quad (85)$$

and

$$b^m b^{n+1} \equiv 1 \text{ Mod } (b-1). \quad (86)$$

Then the check for the diminished radix complement is

$$p_X \oplus p_{\bar{X}} = 0 \quad (87)$$

and the check for the radix complement is

$$p_X \oplus p_{X'} = 1. \quad (88)$$

Congruences are multiplicative. Therefore, a check of the product

$$P = R \times S$$

is given simply as

$$p_P = p_R \odot P_S \quad (89)$$

with no cognizance of carries required.

A check involving the division of congruences is not practical since the correct division procedure is dependent upon whether the divisor and $(b-1)$ have any common factors. However, this is of little consequence since machine division is usually obtained by a process consisting of repetitive add and shift operations.

$$\frac{X}{Y} = Q + \frac{R}{Y}. \quad (90)$$

$$X = QY + R. \quad (91)$$

The check of the division process is independent of carries and must be true at every step in the division process. The check is

$$p_X = (p_Q \odot p_Y) \oplus p_R. \quad (92)$$

### The Augmented Base Numerical Check

The diminished base numerical check avoids the need for carry cognizance and for that reason offers a simpler check procedure than is obtainable by means of digital checking.

However, for the binary number system the diminished base check is meaningless. One solution is the augmented base check. This system is also carry independent, and parity check digits are relatively easy to obtain. The augmented base check is dependent on the following property of congruences.

$$\sigma_i b^j \equiv + \sigma_i \text{ Mod } (b+1) \text{ if } j \text{ is even} \quad (93)$$

$$\equiv - \sigma_i \text{ Mod } (b+1) \text{ if } j \text{ is odd.} \quad (94)$$

The check digit for representation $A$ is given as

$$(\cdots + a_5 - a_4 + a_3 - a_2 + a_1) \equiv p \text{ Mod } (b+1) \quad (95)$$

where

$$A = \cdots + a_5 b^4 + a_4 b^3 + a_3 b^2 + a_2 b^1 + a_1 b^0. \quad (96)$$

The checking procedures employed for the arithmetic processes are similar to the procedures for the diminished radix except that ring addition $\oplus$ and ring multiplication $\odot$ are defined modulo $(b+1)$.

In the binary case the number base is two and the check base is three. The parity of a representation $A$ is given as

$$(\cdots + a_5 + 2a_4 + a_3 + 2a_2 + a_1) \equiv p \text{ Mod } 3 \quad (97)$$

or

$$\cdots \oplus a_5 \oplus 2a_4 \oplus a_3 \oplus 2a_2 \oplus a_1 = p. \quad (98)$$

Direct subtraction is avoided by using the radix complement of one with respect to the base three.

Note that for the binary case the augmented base check is identical to the diminished base check, base four.

### Examples of the Augmented Base Check

Let $b=2$; then the check is modulo 3.

Addition $X+Y=S$

0 0 0 1 1 1 0 1 ⑩
0 1 0 1 0 1 0 1 ⑪
0 1 1 1 0 0 1 0 ⑩⑩

Check
$p_X \oplus p_Y = p_S$
$10 \oplus 01 = 00$

Complement
$A = 00011101.$ ⑩

$n=7$

Diminished Radix Complement
$\overline{A} = 11100010.$ ⑪

Check
$P_A \oplus P_{\overline{A}}$
$= (b^{n+1}-1) \text{ Mod } (b+1)$
since $n+1$ is even
$10 \oplus 01 = 00$

Radix Complement
$A' = 11100011.$ ⑩

Check
$p_A \oplus p_{A'} = b^{n+1} \text{ Mod } (b+1)$
since $n+1$ is even
$10 \oplus 10 = 01$

Multiplication $R \times S = P$

1011 ⑩
×1001 ⑩⑩
1011
101100
1100011 ⑩⑩

Check
$p_R \odot p_S = p_P$
$10 \odot 00 = 00$

Division $X/Y = Q + V_R/Y$

0 110011 $-X$
1 0101 $-Y$
0 000111 $-R$
↑ sign column

Check
$p_X = (p_Q \odot p_Y) \oplus p_R$
$00 = (01 \odot 10) \oplus 01$
$= 10 \oplus 10 = 00$

The above is a check of the first step in the division process.

## Conclusion

The congruence notation used provides a fundamental basis for the concept of both the digital and the numerical type of parity check. Congruence notation also provides the mathematical tools needed to consider the arithmetic properties of the parity checks and the extension of the parity check concept to nonbinary systems.

In the final analysis, the effectiveness of a parity check is dependent upon the error structure of the carry process. It is necessary to consider in detail the mechanism of binary addition before the effectiveness of the parity check may be evaluated. A subsequent paper by the author considers the error structure of the binary adder and the effectiveness of the various parity checks.

# Investigations of Magnetic Amplifiers with Feedback*

HARRY J. GRAY, JR.†

*Summary*—**Sine wave carrier excited magnetic amplifiers have been investigated to determine if the figure of merit can be improved through the use of feedback techniques. It is shown that the power gain can be made unlimited but that a finite rise time is preserved. Hence the figure of merit as ordinarily defined becomes meaningless. It is shown, however, that voltage gain divided by rise time remains nearly constant under feedback and is a more suitably defined figure of merit.**

## Introduction

MAGNETIC amplifiers are attractive devices for computer applications because of their potentially unlimited life, high reliability, and relative insensitivity to environment, when compared with electron tubes. Carrier magnetic amplifiers have not been used in megacycle computer applications in the past because of limited power gains and speed. However, it has been suggested that feedback techniques might improve the power gain with little loss of speed.

There are essentially two possibilities: application of sufficient positive feedback to produce a regenerative pulse amplifier, and use of feedback techniques to produce a stable but fast-rise high-gain amplifier having many potential applications. This paper deals with the problem of understanding the behavior of amplifiers of the second type. The first type is a special case of the second.

## A Heuristic Approach

Consider the self-saturating magnetic amplifier shown

* Manuscript received by the PGEC, March 14, 1958. This work was performed while the author was at the Remington Rand Univac Div. of Sperry-Rand Corp. and was supported by the AF Cambridge Res. Ctr., Air Res. and Dev. Command, under Contract AF19(604)-1376.

† Moore School of Elec. Eng., University of Pennsylvania, Philadelphia, Pa. Formerly at Remington Rand Univac Div. of Sperry-Rand Corp., Philadelphia, Pa.

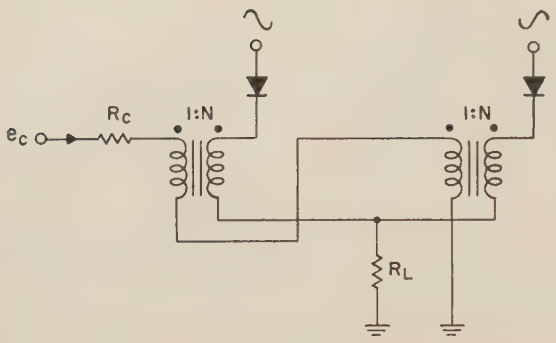



Fig. 1—Self-saturating magnetic amplifier.

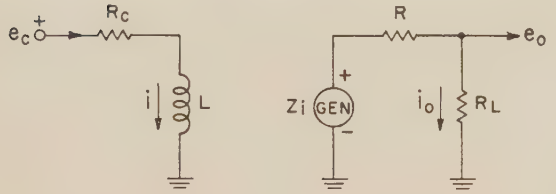



Fig. 2—Equivalent circuits for self-saturating magnetic amplifier.

in Fig. 1. Pulse tests indicate that the terminal behavior is approximately equivalent to the terminal behavior of the circuit seen in Fig. 2. ($Z$ is resistive.)

Consider the application of pure negative voltage feedback and pure positive current feedback to the equivalent circuit of Fig. 2. Practical considerations are dealt with later. Then the equivalent circuit becomes as shown in Fig. 3.

The transformed input impedance is

$$Z_c(s) = R_c + \frac{sL + \dfrac{K_{fv}R_LZ}{R + R_L}}{1 - \dfrac{K_{fi}Z}{R + R_L}}. \qquad (1)$$

The transformed voltage gain is

$$K_v(s) = \frac{\dfrac{ZR_L}{R + R_L}}{R_c\left[1 - \dfrac{K_{fi}Z}{R + R_L}\right] + sL + \dfrac{K_{fv}R_LZ}{R + R_L}}. \qquad (2)$$

The power gain is given by

$$K_p = \frac{Z_c}{R_L} K_v^2$$

with $s=0$. The power gain is infinite when $Z_c$ or $K_v$ is infinite. Examination of the conditions for which $K_v$ is infinite shows that $Z_c$ is then zero. In the limit, $K_p$ is infinite. This corresponds to a current actuated amplifier. From (1), $Z_c$ is infinite when

$$K_{fi} = \frac{R + R_L}{Z}, \qquad (3)$$

that is, when the current feedback is adjusted to the point of instability in the absence of negative voltage feedback. Using this result in (2) yields for the voltage gain and 10–90 per cent rise time, $K_v = 1/K_{fv}$ and

$$T_r = 2.2\,\frac{L(R + R_L)}{K_{fv}ZR_L}.$$

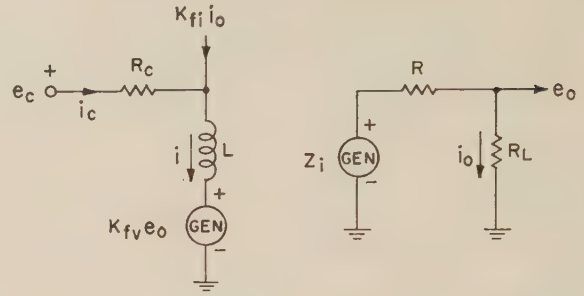



Fig. 3—Equivalent circuits with pure negative voltage feedback and pure positive current feedback applied.

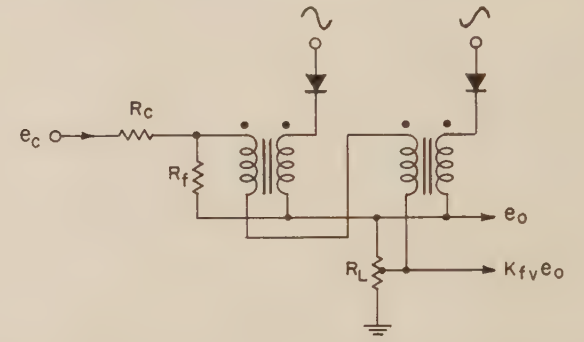



Fig. 4—Magnetic amplifier with positive and negative feedback.

The quantity $K_v/T_r$ becomes

$$K_v/T_r = \frac{ZR_L}{2.2L(R + R_L)} \qquad (4)$$

which is a constant of the amplifier. From (2) it is seen that (4) is valid also for values of $K_{fi}$ other than given by (3).

This last case is similar to vacuum-tube operation. Hence, at present, it is the most interesting case.

The following amplifier was tested in the circuit of of Fig. 4.

Cores = 4 wraps of $\frac{1}{8}$ mil by 1/32 inch 4–79 Mo-Permalloy on 0.1-inch-diameter form.
Output winding = 111 T No. 46 HF.
Control winding = 111 T No. 46 HF (links both cores).
Diodes = IN64.
$R_L = 24\Omega$.
$R_c = 100\Omega$.
RF supply = 12-volt peak at 2.25 mc.

Some early data are summarized in Table I for 0.1-volt pulse input. The voltage gain for $R_f = 470\Omega$ (approximate point of infinite input impedance) was 25 and 5.8 for $K_{fv}$ equal to 0.02 and 0.2, respectively. The agreement with theory is sufficiently good to lend weight to the argument.

TABLE I
$K_v/T_r(\mu s^{-1})$

| $K_{fv}$ \ $R_f$ | 470 | 560 | 680 | 820 | 1000 | 1200 | 1500 |
|---|---|---|---|---|---|---|---|
| 0.02 | 2.6 | 2.0 | 1.7 | 1.7 | 1.8 | 1.5 | 1.6 |
| 0.2 | 1.2 | 1.0 | 1.7 | 1.5 | | | |

The implications of (4) are that the rise time has a limit of zero as $K_v$ approaches zero. This evidently is not true with a finite RF supply frequency. Furthermore the analysis does not give the insight into the performance of the amplifier that is necessary for design. A more rigorous approach requires a cycle-by-cycle analysis in the manner of Johannessen[1] whose remarkable paper suggested some of the items discussed here.

## Difference Equation Analysis

The reader is referred to the Johannessen paper for the mechanics of the analysis. The same assumptions are made here as in that paper, except that:

1) The reactors are assumed to have a finite magnetizing impedance $Z_m$ which is essentially resistive in the megacycle range.
2) The winding resistance is negligible. However, a loss of output volt-seconds occurs due to diode drops and winding reactance when the core is saturated. The effect is assumed to be equivalent to the loss of volt-seconds caused by a fictitious winding resistance, $R_w$.

### *No External Feedback*

A nodal analysis of the circuit of Fig. 1 yields

$$E_0(n+1) = K_{vo}E_c(n) + K_fE_0(n) \tag{5}$$

where

$$K_{vo} = N\frac{R_L}{R_L+R_w}\frac{Z}{N^2R_c+Z}; \qquad Z = \frac{Z_mR_x}{Z_m+R_x}$$

$$K_f = \frac{Z_m}{Z_m+N^2R_c\|R_x} - \frac{Z_m\|N^2R_c}{R_x+Z_m\|N^2R_c}\frac{2R_L+R_w}{R_L+R_w}. \tag{6}$$

The solution of the difference equation (5) for a step in $e_c$ of amplitude $E_c$ applied at the start of interval zero is

$$E_0(n) = \frac{K_{vo}E_c}{1-K_f}[1-K_f{}^n]. \tag{7}$$

Avoiding the quagmire of precise definition, one obtains from (7) as a measure of the rise time and maximum voltage gain

$$T_r = \frac{\pi}{\omega}\left[1+\frac{2.2}{\ln 1/K_f}\right] = \frac{n\pi}{\omega}$$

$$K_v = \frac{K_{vo}}{1-K_f}. \tag{8}$$

The ratio $K_v/T_r$ becomes

$$\frac{K_v}{T_r} = \frac{\omega K_{vo}}{\pi(1-K_f)\left(1+\dfrac{2.2}{\ln 1/K_f}\right)} \tag{9}$$

which is not independent of $K_f$ as in the above section.

However, the quantity

$$\frac{K_v}{T_r}\cdot\frac{\pi}{\omega K_{vo}}$$

is plotted vs $K_f$ and $n$ in Fig. 5, demonstrating that this quantity is fairly constant over a wide range.

Hence for $K_f>0$, a good approximation is

$$\frac{K_v}{T_r} = \frac{\omega K_{vo}}{2\pi} = fk_{vo}. \tag{10}$$

Thus it is reasonable to take $fK_{vo}$ as a figure of merit.

### *Properties of $K_{vo}$*

Eq. (5) describes the behavior of an amplifier which possesses a single time constant; hence it is the simplest to analyze. Internal feedback is present as shown by the term $K_fE_0(n)$. All amplifiers discussed in this paper can be described by (5). External feedback will be used to control $K_f$ and modify the input impedance.

Eq. (10) is a direct and fundamental consequence of (5) giving the figure of merit, $fK_{vo}$, some general significance. From (6) the reciprocal of $K_{vo}$ becomes

$$\frac{1}{K_{vo}} = \frac{1}{N}\left(1+\frac{R_w}{R_L}\right)\left(1+\frac{N^2R_c}{Z}\right),$$

a linear function of $R_c$ or $R_w$. This result allows an experimental check to be made. The amplifier described in the preceding section was operated with a 10-volt peak RF supply at 2.25 mc and with a 100–ohm load. Measurements of voltage gain and rise time were made and $K_{vo}$ computed from (8). The results are plotted vs $R_c$ in Fig. 6. The amplifier was adjusted in each case for maximum voltage gain with a peak filtered output of $\frac{1}{2}$ volt corresponding to a signal input utilizing fully the dynamic range of the amplifier.

The linear dependence on $R_c$ is evident and the slope is consistent with theory if a finite diode reverse impedance ($R_x$) is assumed. However, characteristically of all data of this type, the intercept is too low. The proper intercept is obtained when the amplifier is adjusted for maximum voltage gain with a small-signal input. Plots of this type were made in order to determine experimentally the maximum value of $K_{vo}(K_{vom})$ for an amplifier to compare various amplifier structures before feedback is applied.

[1] P. R. Johannessen, "Analysis of magnetic amplifiers by the use of difference equations," *Trans. AIEE (Commun. and Electronics)*, vol. 73, pp. 700–711; January, 1955.

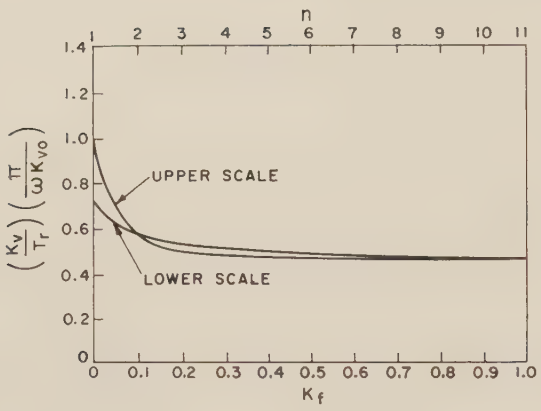



Fig. 5—$(K_v/T_r)(\pi/\omega K_{vo})$ vs $K_f$.

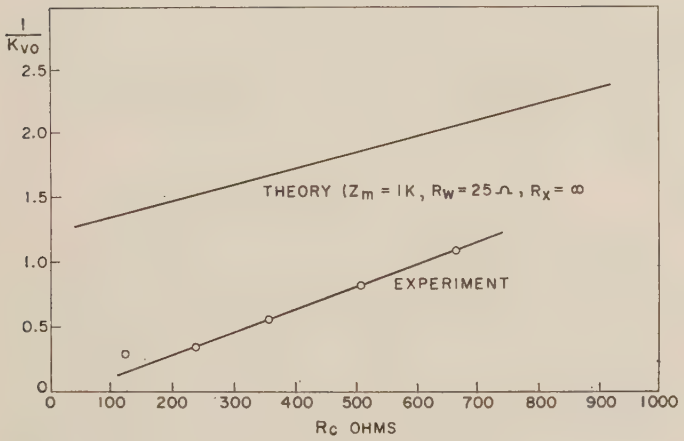



Fig. 6—$1/K_{vo}$ vs $R_c$.

From (6) $K_{vom}=N$, and from (9) the maximum value of $K_v/T_r$ is $N\omega/\pi$. This result can be derived more directly as follows. The input volt-seconds are transformed by the control to output turns ratio, $N$. Hence, neglecting losses, the maximum voltage gain is $N$. The output occurs in one-half cycle, $\pi/\omega$. Taking this to be the rise time, the maximum figure of merit $N\omega/\pi$ is yielded. In a like manner, a single-core amplifier has a maximum voltage gain of $N$, a rise time of one cycle, and therefore a figure of merit, $N\omega/2\pi$.

*External Feedback*

Consider the circuit shown in Fig. 7. A nodal analysis yields for $K_{vo}$ and $K_f$,

$$K_{vo} = N\frac{R_L}{R_L+R_w}\cdot\frac{Z}{N^2R_c}$$

$$K_f = \frac{Z}{N^2R_c\left\|R_x\right\|\frac{N}{M}R_f} - \frac{R_L}{R_L+R_w}\cdot\frac{Z}{\frac{N^2R_c}{NK_{fv}}\left\|\left(-\frac{N}{M}R_f\right)\right\|\frac{R_L}{2R_L+R_w}R_x} \qquad (11)$$

where

$$Z = N^2R_c\left\|Z_m\right\|R_x\left\|\left(\frac{N}{M}\right)^2R_f\right.\text{.}$$

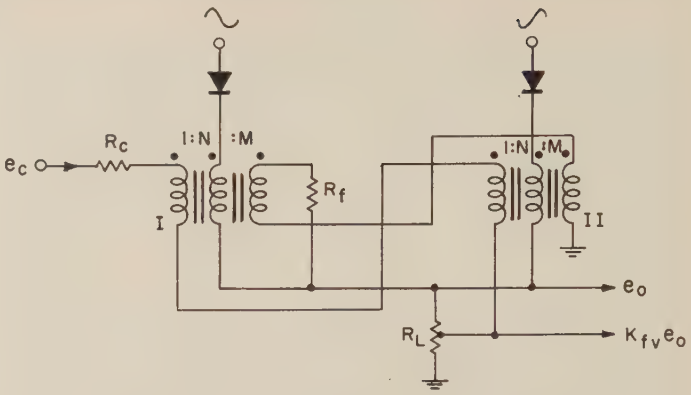



Fig. 7—Magnetic amplifier with feedback winding.

The input impedance is, statically,

$$Z_c = \frac{R_c}{1-K_vK_{fv}}\text{.} \qquad (12)$$

Eq. (12) is sufficiently startling to require an outline of its derivation. Fig. 8 shows equivalent circuits for Fig. 7 for two cases.

Solving for

$$i_{1/N} = \frac{Ne_c+e_I-e_s}{N_2R_c} \qquad \text{cores unsaturated}$$

$$i_{1/N} = \frac{Ne_c+e_I-NK_{fv}}{N^2R_c} \qquad \text{core II saturated.}$$

Averaging the currents over a cycle yields (12).

Note that the input impedance is infinite when $K_{fv}=1/K_v$. To determine the necessary circuit conditions, set $K_{fv}=0$ in the second part of (11) and set $K_f=1$. The resulting equation

$$\frac{Z}{N^2R_c\left\|R_x\right\|\frac{N}{M}R_f} - \frac{R_L}{R_L+R_w}\cdot\frac{Z}{-\frac{N}{M}R_f\left\|\frac{R_L}{2R_L+R_w}\right\|R_x} = 1 \qquad (13)$$

is a constraint on the circuit constants. Calculation of $1-K_f$ subject to the constraint of (13) yields

$$1-K_f = \frac{R_L}{R_L+R_w}Z\frac{NK_{fv}}{N^2R_c} \qquad (14)$$

and

$$K_v = \frac{K_{vo}}{1-K_f} = \frac{1}{K_{fv}}$$

follows from the first part of (11). Hence, when (13) is satisfied, the input impedance is infinite and $K_v=1/K_{fv}$.

The following amplifier was constructed.

Cores = 3 wraps of $\frac{1}{8}$ mil × 1/32 inch 4-79 Mo-Permalloy on 0.1-inch-diameter form.

Power winding = 147T No. 46 HF.

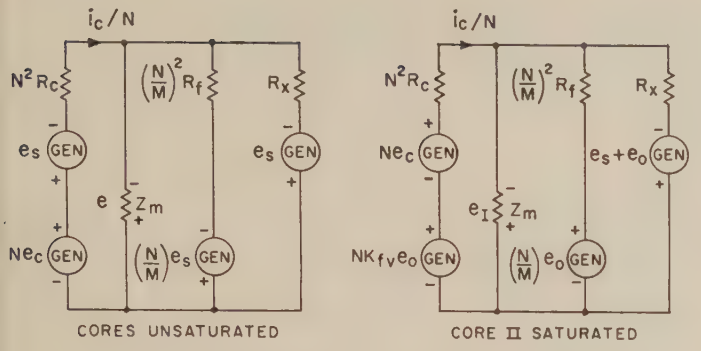



Fig. 8—Equivalent circuits for magnetic amplifier with feedback winding.

Control and feedback windings =

2—147T No. 46 HF
1—74T No. 46 HF
1—37T No. 46 HF
1—18T No. 46 HF.

Diodes = EM15 (high conductance, fast recovery type).

Carrier = 10-volt peak at 2.25 mc.

$R_w = 50\Omega$ by power output vs $R_L$ test.

Various combinations of turns for the feedback and control windings were tried along with various values of $K_{fv}$. Voltage gains, power gains, and rise times were measured for a step input; rise times were estimated by cycle counting.

Typical data with $R_f = 2.2K$, $N = 2$, $M = 4$, $R_L = 100\Omega$ and $Z_c \approx \infty$ are shown in Table II.

TABLE II

| $R_c$ | $K_{fv}$ | $K_v$ | $T_r$(μsec) | $K_v/T_r$ | $K_{vo}$ |
|---|---|---|---|---|---|
| 100 | 0.5 | 2 | 1.34 | 1.49 | 0.66 |
| 470 | 0.5 | 2 | 2.9 | 0.69 | 0.31 |
| 1K | 0.5 | 2 | 4.25 | 0.47 | 0.21 |
| 1K | 0.7 | 1.45 | 3.8 | 0.38 | 0.17 |
| 1K | 1.0 | 1.0 | 2.45 | 0.41 | 0.18 |

The relation $K_v = 1/K_{fv}$ is seen to be satisfied by the data. Note that if $R_c$ is fixed, $K_v/T_r$ is very nearly constant. Also note the dependence of $K_{vo}$ on $R_c$.

That $K_{vo}$ decreases when $R_c$ increases is a severe practical handicap because a driving source having little power output will have a high internal impedance contributing to $R_c$ and lowering $K_{vo}$, hence reducing the voltage gain bandwidth. Empiricism has shown that the addition of frequency sensitive circuit elements in the output and feedback circuits limits the reduction of $K_{vo}$ as $R_c$ is increased. Such measures when applied to this amplifier yielded a voltage gain bandwidth as defined by (10) of 1.8 mc. which was independent of the turns ratios for $1.1 < M < 2.7$ and surprisingly for $0.7 < N < 1.6$. The value of $R_c$ in these tests was $4.7K$. The dependence on $R_c$ was slight.

## Conclusions

It has been demonstrated that feedback techniques can be applied successfully to carrier magnetic amplifiers to obtain useful results. It is shown that power gain bandwidth is not as meaningful as a figure of merit as is voltage gain bandwidth. Furthermore, an amplifier subjected to combined positive and negative feedback can be adjusted to have an infinite input impedance. Under these circumstances, its voltage gain is equal to the reciprocal of the voltage feedback factor and its rise time is given by $K_v/fK_{vo}$.

## Definitions of Symbols

$e_c$ = instantaneous control voltage.
$e_s$ = instantaneous supply voltage.
$e_0$ = instantaneous output voltage.
$e_I$ = instantaneous voltage across core I.
$E$ = average voltages, averaged over a half-cycle of the supply voltage. The same subscripts as for instantaneous voltages are used.
$f$ = frequency of supply voltage, cycles per second $\omega = 2\pi f$.
$i$ = instantaneous current.
$i_c$ = instantaneous control circuit current.
$K_f$ = feedback factor.
$K_{fv}$ = voltage feedback factor.
$K_{fi}$ = current feedback factor.
$K_v$ = voltage gain.
$K_p$ = power gain.
$K_{vo}$ = voltage gain before feedback.
$K_{vom}$ = maximum value of $K_{vo}$.
$M$ = turns ratio of feedback winding to control winding.
$N$ = turns ratio of output winding to control winding.
$R_c$ = resistance of control circuit.
$R_f$ = feedback resistor.
$R_L$ = resistance of load.
$R_w$ = equivalent resistance of output winding.
$R_x$ = equivalent leakage resistance of diode.
$s$ = complex frequency variable.
$T_r$ = rise time of output.
$Z_c$ = input impedance.
$Z_m$ = magnetizing impedance of reactor referred to output winding.
$Z$ = an impedance.
$n-1, n, n+1$ = three consecutive half-cycles of the supply voltage.
$\|$ = an operator enabling impedances instead of admittances to be written in a nodal analysis. Note that

$$R_1 \| R_2 = \frac{R_1 R_2}{R_1 + R_2} = \frac{1}{G_1 + G_2}.$$

Also,

$$R_1 \| - R_2 = -(-R_1 \| R_2)$$

and

$$R_1 \| R_2 \| - R_2 = R_1.$$

# A New Class of Digital Division Methods*

JAMES E. ROBERTSON†

*Summary*—**This paper describes a class of division methods best suited for use in digital computers with facilities for floating point arithmetic. The division methods may be contrasted with conventional division procedures by considering the nature of each quotient digit as generated during the division process. In restoring division, each quotient digit has one of the values 0, 1, · · · , $r-1$, for an arbitrary integer radix $r$. In nonrestoring division, each quotient digit has one of the values $-(r-1)$, · · · , $-1$, $+1$, · · · , $+(r-1)$. For the division methods described here, each quotient digit has one of the values $-n$, $-(n-1)$, · · · , $-1, 0, 1$, · · · , $n-1$, $n$, where $n$ is an integer such that $\frac{1}{2}(r-1) \leq n \leq r-1$. A method for serial conversion of the quotient digits to conventional (restoring) form is given. Examples of new division procedures for radix 4 and radix 10 are given.**

## Introduction[1]

A DIVISION method can be categorized by listing the permissible values of each quotient digit as generated during the division process. For an arbitrary radix $r$, each quotient digit generated during a conventional restoring division has one of the values $0, 1, \cdots, r-1$. For nonrestoring division each quotient digit has one of the values $-(r-1), -(r-2), \cdots, -1, 1, 2, \cdots, r-1$, with 0 excluded. The purpose of this paper is to describe a class of division methods in which each quotient digit has one of the values $-n, -(n-1), \cdots, -1, 0, 1, \cdots, n$, where $n$ is an integer such that $\frac{1}{2}(r-1) < n < r-1$. Conversion of the quotient to the conventional restoring form is required for the latter two classes of division methods.

Division, as executed in most digital computers now in use, involves a recursive process which may be preceded by preliminary operations and followed by terminal operations. Most of the time required for a digital division is spent in the repeated execution of the recursive process. For nonrestoring division and for the class of division methods proposed here, the recursive process can be described by

$$x_{j+1} = rx_j - q_{j+1}d \qquad j = 0, 1, \cdots, m-1$$

for which the following notation is employed:

$x_j$ = partial remainder resulting from the $j$th execution of the recursive process
$x_0$ = dividend
$x_m$ = remainder
$q_j$ = (for fractions) the $j$th digit of the quotient to the right of the radix point
$m$ = the number of digits, radix $r$, used to represent the quotient
$d$ = divisor.

Each $x_j (j = 0, 1, \cdots, m)$ satisfies $|x_j| \leq k|d|$, and the sign of each $q_{j+1}$ is chosen so that $|x_{j+1}| = |(r|x_j| - |q_{j+1}||d|)|$. A straightforward but lengthy analysis of all cases that may arise reveals another property of each quotient digit, namely $|q_{j+1}| \leq k(r-1)$. For nonrestoring division, $k = 1$; it is shown that division methods exist for certain discrete values of $k$ in the range $\frac{1}{2} \leq k \leq 1$.

From the equation for the recursive process it can be shown readily that the division procedure is correct.

For $j = 0$,

$$x_1 = rx_0 - q_1 d.$$

For $j = 1$,

$$x_2 = rx_1 - q_2 d = r^2 x_0 - (rq_1 + q_2)d.$$

For $j = m-1$,

$$x_m = r^m x_0 - (r^{m-1}q_1 + r^{m-2}q_2 + \cdots + rq_{m-1} + q_m)d.$$

The shifted remainder $r^{-m}x_m$ is then

$$r^{-m}x_m = x_0 - d\sum_{i=1}^{m} r^{-i}q_i$$

where

$$\sum_{i=1}^{m} r^{-i}q_i$$

represents the quotient $Q$. It follows that $Qd + r^{-m}x_m = x_0$; *i.e.*, the sum of the shifted remainder and the product of the quotient and divisor is the dividend.

The mechanization of the recursive process requires three distinct steps.

1) The partial remainder $x_j$ is shifted, *i.e.*, multiplied by the radix $r$.
2) One of several permissible arithmetic procedures is selected, such that the maximum absolute value of $r|x_j|$, namely $kr|d|$, is reduced by the amount $1/r$, so that the result $x_{j+1}$ satisfies $|x_{j+1}| \leq k|d|$. It should be emphasized that the reduction is in the range over which partial remainders may vary, and not necessarily in the absolute values of specific partial remainders.
3) A quotient digit is generated corresponding to the arithmetic procedure selected.

* Manuscript received by the PGEC, March 19, 1958; revised manuscript received, July 1, 1958. The investigations leading to this paper were part of a computer study program by the staff of the Digital Computer Lab., University of Illinois, Urbana, Ill., under the joint support of the Office of Naval Res. and the Atomic Energy Comm.

† University of Illinois, Urbana, Ill.

[1] Descriptions of conventional restoring and nonrestoring methods can be found in a number of texts, such as the following.

R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., New York, N. Y.; 1955.

M. Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1958.

The key to the study of division methods lies in the analysis of arithmetic procedures which reduce the allowable range of absolute values of the shifted partial remainder ($rx_j$) by the amount $1/r$.

## Analysis of Arithmetic Procedures

It is convenient to normalize the partial remainders with respect to the absolute value of the divisor. If the substitution $z_j = x_j/|d|$ is made, the range restrictions become $-k \leq z_{j+1} \leq k$ and $-rk \leq rz_j \leq rk$. Attention then is focused on arithmetic procedures which transform $rz_j$ into $z_{j+1}$. The easily mechanized procedures involve addition or subtraction of integral multiples of the divisor from $rx_j$ to yield $x_{j+1}$; after normalization, the procedures involve addition or subtraction of integers from $rz_j$ to yield $z_{j+1}$. The arithmetic procedures can be represented as a family of straight lines of the form $z_{j+1} = rz_j - i$, where $i = -n, \cdots, -1, 0, 1, 2, \cdots, n$, as shown in Fig. 1.

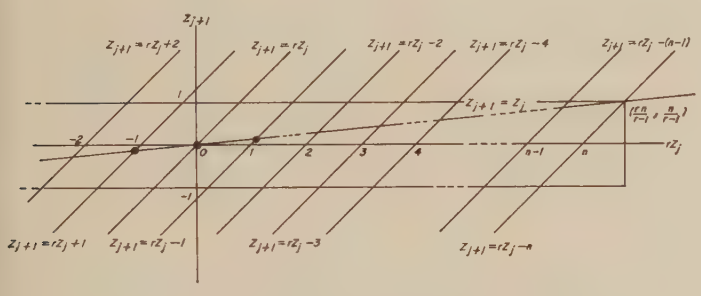



Fig. 1—A normalized graph illustrating arithmetic procedures during division.

In order for one of the proposed class of division methods to exist, it must be possible to superimpose a rectangle on the family of straight lines of Fig. 1 in such a way that the following occur.

1) The rectangle is centered at the origin, with vertices at $(\pm rk, \pm k)$.
2) The projections on the $rz_j$ axis of the line segments within the rectangle cover that portion of the $rz_j$ axis within the rectangle.

It follows from the first condition that the vertices of the rectangle lie on the lines through the origin of slope $\pm 1/r$, whose equations are $z_{j+1} = \pm z_j$. The second condition requires that the rectangle be sufficiently large to insure that $k \geq \frac{1}{2}$. Two additional considerations govern the choice of size of the rectangle, *i.e.*, the choice of $k$ and of $n$.

3) The number of lines of the form $z_{j+1} = rz_j - i$ necessary to satisfy condition 2) should be minimized. The number of multiples of the divisor which must be formed is proportional to the number of lines employed.
4) The overlap of projections of the line segments on the $rz_j$ axis should be maximized. The precision necessary in the selection process decreases as the overlap increases.

The two considerations are mutually contradictory since 3) requires that the size of the rectangle should be decreased and 4) requires that the size be increased.

The considerations govern the choice of $k$ to the extent that $k$ should take one of a discrete set of values such that the vertex $(rk, k)$ of the rectangle lies on a line $z_{j+1} = rz_j - n$. Any one division method can then be characterized by the positive integers chosen for $r$ and for $n$.

The value of $k$ as a function of $r$ and $n$ can be found by solving for the point of intersection of the lines $z_{j+1} = z_j$ and $z_{j+1} = rz_j - n$. The value of $z_{j+1}$ at the point of intersection is $k$, and is found to be $n/(r-1)$. The requirement that $k \geq \frac{1}{2}$ becomes $n \geq (r-1)/2$.

The choice of $n$ for some given radix $r$ involves a balance between time and equipment costs associated with the selection process, on the one hand, and similar costs in forming multiples of the divisor, on the other. Since the balance is so much a function of design details, the choice of $n$ is discussed further only in connection with specific examples.

## Quotient Conversion

The conventional representation of a quotient requires that each digit be one of the positive integers $0, 1, \cdots, r-1$. Since nonrestoring division and the method described here involve negative digits in the quotient, some means of conversion is required. The technique employed in conventional nonrestoring division, except for the special case of the binary system, can be described for a radix complement representation by the following rules.

1) If $q_1 < 0$, replace $q_1$ by $q_1' = r + q_1$ and set the sign of the quotient negative; if $q_1 > 0$, $q_1' = q_1$, and the quotient is positive.
2) For $j = 1, 2, \cdots, m-1$, inspect $q_j'$ and $q_{j+1}$. If $q_{j+1} < 0$, replace $q_j'$ by $q_j' - 1$ and replace $q_{j+1}$ by $q_{j+1}' = r + q_{j+1}$. If $q_{j+1} > 0$, $q_j'$ is left unchanged, and $q_{j+1}' = q_{j+1}$.

The rules require a serial inspection of the quotient digits, the most significant digit first. When a negative digit is encountered, it is added to the radix, and a unit is borrowed from the next most significant digit.

For the proposed division method, the conversion is complicated by the fact that 0's are permissible quotient digits. The inspection of the sign of $q_{j+1}$ (or $q_1$ in rule 1) must be replaced by an inspection of signs of the divisor $d$ and the partial remainder $x_j$ ($x_0$ in rule 1) to determine the sign of the next *nonzero* quotient digit. Agreement of signs of $x_j$ and $d$ corresponds to $q_{j+1} > 0$ in the above rules; disagreement corresponds to $q_{j+1} < 0$. Alternatively, signs can be determined in the usual way and can be associated with those $q_{j+1}$ which are zero. These modifications provide for a borrow propagation through a sequence of 0's.

Both sets of conversion rules require that no quotient digit $q_j$ be such that $|q_j| > r-1$. In particular, the

parameter $n$ is one value that $|q_j|$ can assume, and therefore $n \leq r-1$. Thus, $n$ is restricted to the range $\frac{1}{2}(r-1) \leq n \leq r-1$, since it was established previously that $n \geq \frac{1}{2}(r-1)$.

### Preliminary and Terminal Operations

Some of the requirements which necessitate preliminary or terminal operations for division methods are listed below.

1) The requirement of standardizing the quotient, in a floating point division, and the requirement of overflow detection in a fixed point unit.
2) The requirement for a rounded quotient.
3) The requirement that a correct remainder be generated.

Procedures vary in computers now in use. For the division methods described in this paper, the above requirements necessitate comparable preliminary or terminal operations.

An additional requirement imposed by the division methods discussed is that each partial remainder $x_j$ should satisfy $|x_j| \leq k|d|$ where $d$ is the divisor and $\frac{1}{2} \leq k \leq 1$. In particular, the restriction applies to the dividend $x_0$, and may necessitate additional preliminary operations for division methods for which $k<1$, in contrast to conventional procedures for which $k=1$.

Preliminary standardization of the divisor simplifies the selection of the correct multiple of the divisor during the recursive process, since the precision required for selection increases as the minimum absolute value of the divisor decreases. The proposed methods, therefore, are best suited for use in arithmetic units having facilities for floating point operations. The methods can be used for fixed point division if facilities for simultaneously shifting divisor and dividend are available.

Additional problems are posed when a remainder $x_m$ must be generated such that the division algorithm $Qd+r^{-m}x_m=x_0$ (where $Q$ is the quotient represented by $m$ digits, radix $r$; $d$ is the divisor, and $x_0$ the dividend), is satisfied. If $d$ and $x_0$ are initially shifted left $p$ digital positions so that $d'=r^p d$ and $x_0'=r^p x_0$, then the value of $x_m'$ such that $Qd'+r^{-m}x_m'=x_0'$ is found to be $x_m'=r^p x_m$. For a fixed point division, it would thus be necessary to shift the remainder $x_m'$ $p$ digital positions right to obtain the correct remainder $x_m$.

A second difficulty arises when the quotient conversion rules require the least significant digit $q_m'$ of the converted quotient to have the value $r$. If, as is often the case, facilities for addition are not available for the quotient register, $q_m'$ can be set to the value $r-1$, and the remainder $x_m$ must then be replaced by $x_m'=x_m+d$. The division algorithm becomes

$$(Q - r^{-m})d + r^{-m}(x_m + d) = x_0.$$

Similar results are obtained when conventional nonrestoring division methods are employed.

*Example 1—Radix 4 Division*

Conventional radix 4 division methods require either two uses of a binary adder (use of one adder sequentially or two adders in parallel) or the formation and storage to full precision of $3d$, where $d$ is the divisor. The division method of the class proposed here with $r=4$ and $n=2$ requires a single binary adder, conditional doubling and complementing circuits, and a selection circuit to compare $rx_j$ with $0.5d$ and $1.5d$ to a precision of 7 binary digits, if $\frac{1}{4} \leq |d| \leq 1$.

The mechanization of the division scheme is indicated diagrammatically in Fig. 2.

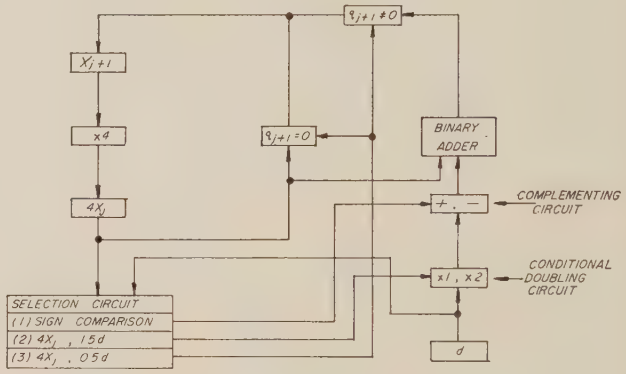



Fig. 2—Radix 4 division.

The selection circuit performs three functions, based upon the seven most significant binary digits of $4x_j$ (or $x_j$ if desired) and of $d$.

1) Compares signs of $x_j$ and $d$. If signs agree, the complementing circuit is set to subtract; if signs disagree, the complementing circuit is set to add.
2) Compares absolute values of $4x_j$ and $1.5d$. If $4|x_j| \geq 1\frac{2}{3}|d|$, the conditional doubling circuit must be set to form $2d$. If $4|x_j| \leq 1\frac{1}{3}|d|$, the circuit must be set to form $d$. If $1\frac{1}{3}|d| < 4|x_j| < 1\frac{2}{3}|d|$, the conditional doubling circuit can be set either way, depending upon the design details of the selection circuit.
3) Makes a similar comparison of absolute values of $4x_j$ and $(\frac{1}{2} \pm \frac{1}{6})d$. For the smaller values of $4|x_j|$, $x_{j+1}=4x_j$; otherwise $x_{j+1}$ is transferred from the binary adder.

After $x_{j+1}$ is formed, $4x_{j+1}$ is formed by a radix 4 left shift to replace $4x_j$. The values selected for the quotient digit $q_{j+1}$ must correspond to the selections made, as summarized in Table I.

TABLE I

| Selection 1 | Selection 2 | Selection 3 | $q_{j+1}$ |
|---|---|---|---|
| Subtract | $2d$ | $q_{j+1} \neq 0$ | $+2$ |
| Subtract | $1d$ | $q_{j+1} = 0$ | $+0$ |
| Subtract | $1d$ | $q_{j+1} \neq 0$ | $+1$ |
| Add | $2d$ | $\neq 0$ | $-2$ |
| Add | $1d$ | $= 0$ | $-0$ |
| Add | $1d$ | $\neq 0$ | $-1$ |

*Example 2—Radix 10 Division*

From the many possible choices available to the designer, the radix 10 division method of this example is based upon the following.

1) The excess three representation of decimal digits is chosen.
2) A single decimal adder (excess three code) is used sequentially.
3) Storage is provided for the divisor $d$, but not for any of its multiples.
4) A complementing circuit and a conditional doubling and quintupling circuit are employed, with $\pm d$, $\pm 2d$, and $\pm 5d$ available as inputs to the adder.
5) The division method is characterized by $r=10$, $n=7$, with

$$k = \frac{n}{r-1} = \frac{7}{9}.$$

Doubling and quintupling circuits can be described by the sets of Boolean equations

$$\left.\begin{aligned} q &= b(a \vee c \vee de) \vee ac(d \vee e) \\ r &= \bar{c} \oplus (\bar{a}de \vee a\bar{d}\bar{e}) \\ s &= a \oplus d \oplus e \\ t &= \bar{e} \\ u &= a \end{aligned}\right\} \text{for doubling}$$

and

$$\left.\begin{aligned} v &= e \\ w &= a(\bar{e} \vee b \vee cd) \vee \bar{e}b(c \vee d) \\ x &= \bar{b} \oplus (ecd \vee \bar{e}\bar{c}\bar{d}) \\ y &= \bar{e} \oplus c \oplus d \\ z &= \bar{d} \end{aligned}\right\} \text{for quintupling}$$

where, in excess three notation, $a$, $b$, $c$, $d$; $q$, $r$, $s$, $t$; and $v$, $w$, $x$, $y$ represent one decimal digit of the divisor, $2x$(divisor), and $5x$(divisor), respectively.

For doubling, $e$ is the binary carry input and $u$ is the binary carry output to the next most significant decimal digit. For quintupling, $e$ and $z$ are most easily described as incoming and outgoing binary borrow signals in a halving circuit, with a wired-in decimal shift of $v$, $w$, $x$, and $y$ converting the halving circuit to a quintupling circuit.

If the permutation $a'=b$, $b'=c$, $c'=d$, $d'=e$, $e'=\bar{a}$ is made in the divisor digits, and the permutation $v=\bar{u}$, $w=q$, $x=r$, $y=s$, $z=t$, at the output of the circuit, the doubling circuit is transformed into a quintupling circuit. Thus the same hardware can be used sequentially for both doubling and quintupling, provided permutation and complementation of input and output signals is correctly arranged.

The arrangement of hardware required for the division scheme is shown in Fig. 3. The two steps required for the generation of each decimal quotient digit can be described as follows.

Each quotient digit $q_{j+1}$ can be decomposed into two digits $q_{j+1}'$, $q_{j+1}''$, such that $q_{j+1}=q_{j+1}'+q_{j+1}''$, where $q_{j+1}'=-5$, 0, or 5 and $q_{j+1}''=-2$, $-1$, 0, 1, or 2. Step 1 corresponds to the determination of $q_{j+1}'$ and results in the formation of a quantity $x_{j+1}'$ such that

$$x_{j+1}' = 10x_j \text{ if } q_{j+1}' = 0$$

or

$$|x_{j+1}'| = |(10|x_j| - 5|d|)| \quad \text{if } q_{j+1}' = \pm 5.$$

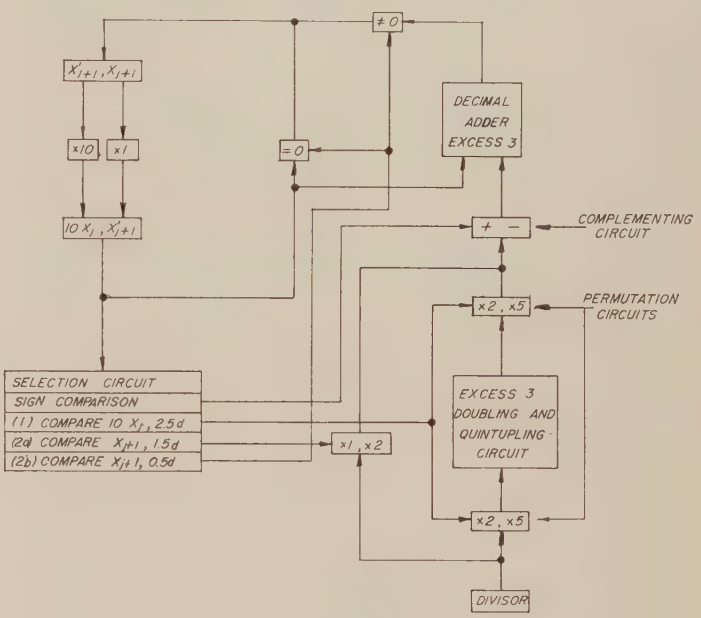



Fig. 3—A decimal division method.

Step 1 reduces the range $0 \le 10|x_j| < 7\frac{7}{9}|d|$ to $0 \le |x_{j+1}'| < 2\frac{7}{9}|d|$. The result of step 2 is $x_{j+1}$, whose range is $0 \le |x_{j+1}| < \frac{7}{9}|d|$. Step 1 is not required whenever $q_{j+1}'=0$. The details of operations performed for all ranges of $10x_j$ and $x_{j+1}'$ are summarized in Table II. The overlap of $\frac{5}{9}|d|$ in the ranges is such that three decimal digits of $d$ and of $10x_j$ or $x_{j+1}'$ are required for

TABLE II

| Range of $10x_j$ | Step 1 | $q_{j+1}'$ | Range of $x_{j+1}'$ |
|---|---|---|---|
| $-7\frac{7}{9}\|d\| < 10x_j < -2\frac{2}{9}\|d\|$ | Add $5\|d\|$ | $-5$ | $-2\frac{7}{9}\|d\| < x_{j+1}' < 2\frac{7}{9}\|d\|$ |
| $-2\frac{7}{9}\|d\| < 10x_j < 2\frac{7}{9}\|d\|$ | Not required | 0 | |
| $2\frac{2}{9}\|d\| < 10x_j < 7\frac{7}{9}\|d\|$ | Subtract $5\|d\|$ | $+5$ | |

| Range of $x_{j+1}'$ | Step 2 | $q_{j+1}''$ | Range of $x_{j+1}$ |
|---|---|---|---|
| $-2\frac{7}{9}\|d\| < x_{j+1}' < -1\frac{2}{9}\|d\|$ | Add $2\|d\|$ | $-2$ | $-\frac{7}{9}\|d\| < x_{j+1} < \frac{7}{9}\|d\|$ |
| $-1\frac{7}{9}\|d\| < x_{j+1}' < -\frac{2}{9}\|d\|$ | Add $\|d\|$ | $-1$ | |
| $-\frac{7}{9}\|d\| < x_{j+1}' < \frac{7}{9}\|d\|$ | | 0 | |
| $\frac{2}{9}\|d\| < x_{j+1}' < 1\frac{7}{9}\|d\|$ | Subtract $\|d\|$ | $+1$ | |
| $1\frac{2}{9}\|d\| < x_{j+1}' < 2\frac{7}{9}\|d\|$ | Subtract $2\|d\|$ | $+2$ | |

comparison, provided $d$ is standardized to the range $1/10 \leq |d| \leq 1$.

The average number of operations necessary for the division method, assuming all quotient digits $-7, \cdots, +7$ are equally likely, is $1\frac{2}{3}$ operations per digit of the quotient. This figure is to be compared with 3.4 operations per quotient digit for a conventional nonrestoring division method employing doubling and quintupling circuits.[2]

## Conclusion

The new methods of division described here, when coupled with two rather obvious comments, show promise of leading to new developments in digital arithmetic. These comments are:

1) Multiplication is the inverse of division.
2) The representation of quotient digits is, at least in the interesting cases, redundant.

This paper is the result of an attempt to find a division inverse to a multiplication method recently described by Lehman.[3] The radix 4 division example is the inverse of the radix 4 equivalent of the binary multiplication described by Lehman, except for the manner in which redundancy is employed in the generation of quoteint digits in the one case and the recoding of multiplier digits in the other. The multiplication method which is the inverse to the radix 10 division is similar to that used on the IBM 602A computing punch;[4] the fact that fewer operations are required per quotient digit for the division method indicates that a better multiplier digit encoding scheme would reduce the number of operations required for multiplication. It is clear that multiplication methods inverse to other division methods of the class described here must exist, and it is felt that further investigation will lead to a broader understanding of digital arithmetic. The interrelationships between quotient digit redundancy, selection procedures, and divisor multiple generation should be studied also, particularly in those cases for which several steps per quotient digit are desirable (*e.g.*, the radix 10 example).

[2] Richards, *op. cit.*, pp. 274–275.
[3] M. Lehman, "High-speed digital multiplication," IRE Trans. on Electronic Computers, vol. EC-6, pp. 204–205; September, 1957.
[4] Richards, *op. cit.*, pp. 262–263.

## Appendix

### Precision Required for Selection

An estimate of the precision required of the divisor $d$ and the shifted partial remainder $rx_j$ for selection of the correct arithmetic procedure can be gained by estimating the variation in the ratio $rx_j/d$ resulting from truncation of $rx_j$ and of $d$. This variation must be less than the overlap of projections on the $rz_j$ axis (Fig. 1) of two successive lines of the form $z_{j+1} = rz_j - i$ $z_{j+1}' = rz_j - (i+1)$.

The overlap is the difference in values of $rz_j$ for $z_{j+1} = n/(r-1)$, and for $z_{j+1}' = -n/(r-1)$, and is $2n/(r-1) - 1$. Truncation errors in $rx_j$ and $d$ can be expressed by setting $a \leq |x_j| \leq a + \Delta a$, $b \leq |d| \leq b + \Delta b$. The variation in the estimates of the ratio $|rx_j/d|$ is then

$$r\left[\frac{a+\Delta a}{b} - \frac{a}{b+\Delta b}\right].$$

It is required therefore that

$$\frac{a+\Delta a}{b} - \frac{a}{b+\Delta b} \simeq \frac{1}{b}\left(\Delta a + \frac{a}{b}\Delta b\right) < \frac{1}{r}\left(\frac{2n}{r-1} - 1\right).$$

Assuming $\Delta a = \Delta b$, one obtains

$$\Delta a < \frac{b[2n - (r-1)]}{(r-1)r(1 + a/b)}.$$

The minimum value of $b$, assuming standardization, radix $r$, is $b = 1/r$; the maximum value of $r(a/b)$ is $\frac{1}{2}(2n-1)$. Therefore,

$$\Delta a < \frac{2[2n - (r-1)]}{r(r-1)(2r + 2n - 1)}.$$

For the example $r = 4$, $n = 2$; $\Delta a < 1/66$, indicating that seven binary digits are required for the comparison. For $r = 10$, $n = 7$; $\Delta a < 1/297$, indicating that three decimal digits are sufficient for the comparison.

## Acknowledgment

The author is grateful for many helpful discussions with staff members of the Digital Computer Laboratory of the University of Illinois.

# Magnetic Core Pulse-Switching Circuits for Standard Packages*

JACK L. ROSENFELD†

*Summary*—A new method for the logical design of magnetic core pulse-switching circuits is presented. This method has features which make it excellent for use in standard packages. These features are the absence of spurious noise signals at the output; the fact that outputs are independent of the order of arrival of input pulses; the fact that interchanging components does not affect circuit behavior; and the fact that moderate changes in clock pulse amplitude and duration do not cause false operation. Furthermore, the number of components required by this system compares favorably with the numbers required by other systems. A computing system can be built by properly interconnecting a few different types of such packages.

The new system uses advance current drive but performs the logical operations with both forward and backward windings in an output network. The use of both types of windings permits a reduction in the total number of cores required.

Tests performed on actual circuits yielded very encouraging results. The circuits operated as predicted, and the performance was most satisfactory.

## Introduction

SEVERAL methods [1]–[8] have been developed for using magnetic cores as logical elements in combinational switching circuits. For certain applications an all-core computing system (except for transistor or vacuum tube driver) may be desired, and to design each individual switching circuit would be a prodigious task for even a small computing system. No present method of constructing magnetic core switching circuits seems to lend itself to economical standardization.

This paper describes a method of magnetic core pulse-switching which enables the logical designer to construct a switching circuit by wiring together a few standard building blocks, using a small number of cores and diodes. The method is quite economical and especially adaptable for application to standard building blocks.

## Analysis of Logical Design

In this paper magnetic core circuits are drawn with modified mirror symbols [5]. The fact that a conductor is wound on a particular core is indicated by a short line segment (the "mirror") cutting the conductor line at a 45° angle (see Fig. 1). This mirror is labeled with the designation of the core on which the winding appears. The number of turns in a winding may be noted beside the mirror symbol ($N_a$, $N_b$, and $N_c$ in Fig. 1). Current through the winding on a core produces flux in either of two directions, depending upon the direction of the winding. This direction is specified in mirror symbols by "reflecting" the current in the mirror. When the direction of flux in a core is changed by current in one of its windings, a voltage appears on the other windings. The polarity of this voltage can be found by reflecting the setting current in its mirror, inverting the direction of the reflection, and reflecting the resultant "ray" in the mirrors for each winding of the core. These secondary reflections show the direction of the induced voltages. Fig. 1 shows magnetic core $A$ set in the "downward" direction by current $I$. If the flux was previously "upward," voltage is induced in the other windings, tending to force current to flow "backward" (from right to left) in lead $l_1$ and "forward" (from left to right) in lead $l_2$. Lead $l_3$ is not connected to a winding on core $A$. Unless otherwise indicated, input currents flow from left to right.

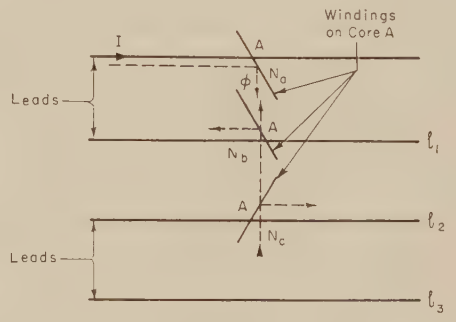



Fig. 1—Modified mirror symbols.

A randomly selected switching function of four variables, $F(x_1, x_2, x_3, x_4) = x_3'x_4 + x_1'x_2x_3'x_4' + x_1'x_2'x_3x_4 + x_1x_2x_3x_4'$, was chosen to illustrate the operation of the new logical system. This is shown in Fig. 2. The pulse sequence is as follows: inputs $x_i$ are pulsed; then advance current $A$ is pulsed, producing output $F$ across load $Z_L$. When the advance lead is pulsed, those cores to which inputs have been previously applied deliver voltages across their output windings. Each of the four leads $l_i$ corresponds to one of the four terms in a disjunctive expression of the demonstration function. Observe that one and only one winding on each lead is wound in the forward direction. The resultant voltage across a lead (from node $a$ to node $b$ for lead $l_1$) can be in the forward direction only if the one forward winding on that lead, and no other winding, is excited. For example, lead $l_1$ will have forward current induced if core $x_3'$ has been set and core $x_4'$ has not been set. If

* Manuscript received by the PGEC, March 20, 1958. This paper is based on the author's thesis of the same title submitted in partial fulfillment of the requirements for the M.S. degree at M.I.T., Dept. of Elec. Eng., June, 1957. The work described was done while the author was employed by Bell Telephone Labs.

† M.I.T., Cambridge, Mass.

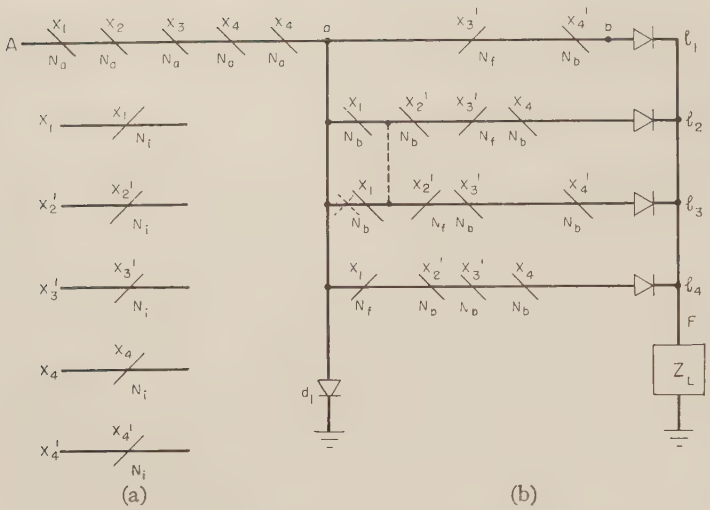



Fig. 2—New logical system. (a) Inputs, (b) output windings.

both cores have been set, their output voltages cancel each other. If core $x_4'$ alone has been set, the back voltage is such as to prevent current from flowing forward in lead $l_1$. Thus this lead corresponds to the term $x_3'x_4$. When this lead, or any other lead, has a forward current induced, the advance current is forced to pass through load $Z_L$. When none of the leads are forward biased, the advance current flows through diode $d_1$. Thus the output function is realized.

These output windings may be combined in series-parallel fashion to eliminate windings. For example, the dashed wire may be soldered in and the dashed cut made in lead $l_3$. This eliminates one of the windings on core $x_1$ and corresponds to the identity $x_1'x_2x_3'x_4'+x_1'x_2'x_3x_4 = x_1'(x_2x_3'x_4'+x_2'x_3x_4)$. Similar modifications may be made as long as every path has one and only one forward winding, as illustrated in Fig. 3. (Manipulations for removing superfluous diodes are discussed in the following section.)

This new system is partly a combination of systems developed by others, combining them in such a manner as to produce decided advantages. The novel feature is the introduction of both forward and backward windings in the output network. The forward windings force current through the load for $F=1$; in this respect the circuit behaves like an $AF$ switch [5]. When $F=0$, the backward windings prevent current from flowing to the load, as in $AB$-type circuits. The advantages of this system will become clear as the discussion proceeds.

## Restrictions on Logical Design

Some of the difficulties encountered in the electrical design of magnetic core switching circuits can most easily be eliminated by placing restrictions on the logical design of the circuitry. In the following discussion it will be assumed that the loads of the core switching circuits are other cores—an appropriate decision for an all-core computing system. But first, some general considerations should be made.

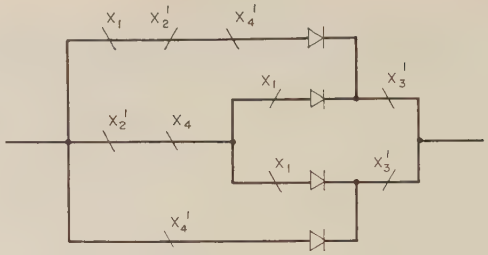
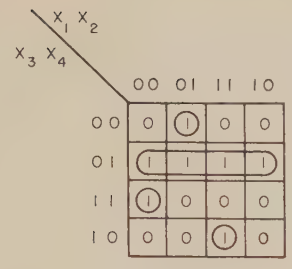


Fig. 3—Output network in minimal form.

The load core of a switching circuit must switch faster than the core which is driving it; if this is not so the driving core will stop switching before the load core is completely switched. When a logic circuit core is driving a load, the resultant drive on the circuit core is $N_aI-N_fI$, since there are $N_aI$ ampere-turns driving the core "down" and $N_fI$ driving it "up." ($N_a$ is the number of turns in an advance winding; $N_f$, forward winding; $N_b$, backward winding; $N_i$, load core input winding; and $I$, the advance current.) Sands [9] has shown that the variation of inverse switching time with magneto-motive force (drive) is linear. Therefore, $N_aI-N_fI$ must be less than the drive to the load core, $N_iI$, if the load core is to switch faster than the driving core. However, if the advance current were to split equally between two paths, the net drive on the driving cores would be $(N_a-\frac{1}{2}N_f)I$. If $N_a-\frac{1}{2}N_f>N_i$, then the driving cores may finish switching before the load core is completely switched, but if no two paths conduct simultaneously this difficulty does not occur.

Thus the switching function to be realized is written in disjunctive form before translating it to circuit form. Each path corresponds to one product term of the sum of disjunctive products, so no two paths may ever conduct simultaneously. In Fig. 2 the paths have been designed to correspond to the disjunctive terms of the demonstration function, as described by the Karnaugh map [10] of Fig. 4.

The above discussion also implies certain restrictions on the design of multiple-output switching circuits. Consider a circuit which is to drive two load cores. Load core 1 is to switch for $F_1(x_1, x_2, x_3)=1$, and load core 2 for $F_2(x_1, x_2, x_3)=1$. If there are some combinations of inputs for which both $F_1$ and $F_2$ are unity, there must be three mutually exclusive (disjunctive) outputs. Output 1 carries a current pulse when load 1 alone is to switch ($f_1=F_1F_2'$); output 2 carries a pulse when load 2 alone is to switch ($f_2=F_1'F_2$); and output 3 is pulsed when both load 1 and load 2 are to switch ($f_3=F_1F_2$). Output 1 is connected to an input winding on load core 1; output 2 is connected to an input winding on load core 2; output 3 is connected to an input winding on load 1 in series with one on load 2. Of course, the circuits for each output must be designed with all paths disjunctive. This restriction on multiple output circuits just described for two load cores may be generalized for any number of load cores. Fig. 5




$$F = X_3' X_4 + X_1' X_2 X_3' X_4' + X_1' X_2' X_3 X_4 + X_1 X_2 X_3 X_4'$$

Fig. 4—Karnaugh map of the switching function in disjunctive form.

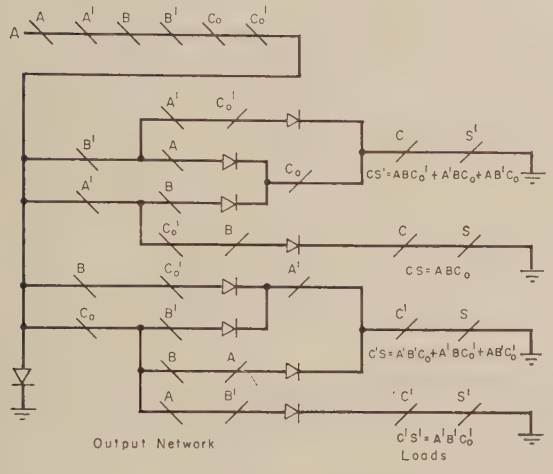



Fig. 5—Output network and loads of binary serial full adder.

illustrates these principles. It is the output network of a binary serial full adder with inputs $A$ and $B$ added to $C_0$, the carry digit from the previous time slot. The outputs are $S$, the sum, and $C$, the carry. The complements of the inputs are furnished, and the circuit delivers the complements of the outputs.

A further restriction upon the design of multiple-output circuits is that no magnetic core may furnish the forward winding for more than one output. Observe that this restriction is satisfied by the binary adder circuit of Fig. 5. Output $CS'$ is driven by forward windings on cores $C_0$ and $C_0'$; $CS$ is driven by $B$; $C'S$ by $A'$ and $A$; and $C'S'$ by $B'$. (Refer now to Fig. 6.) If a driving circuit core $(A)$ should have forward windings ($A_1$ and $A_2$) in the paths of two disjunctive loads, $L_1$ and $L_2$, the following situation might arise: when the inputs are such that $L_1$ should be switched, the forward core $(A)$ in the path of $L_1$ is driven by $(N_a - N_f)I$ ampere-turns; whereas a core with a backward winding $(B')$ in the path of $L_2$ is driven by $N_aI$ ampere-turns. As a result, the "backward" core $(B')$ finishes switching much sooner than the "forward" core $(A)$. If the "forward" core $(A)$ has a forward winding $(A_2)$ in the path of $L_2$, then the forward voltage forces some current to flow in $L_2$ when none should flow, after the "backward" core $(B')$ finishes switching. The circuit of Fig. 5 operates properly because forward windings of different cores drive each output.

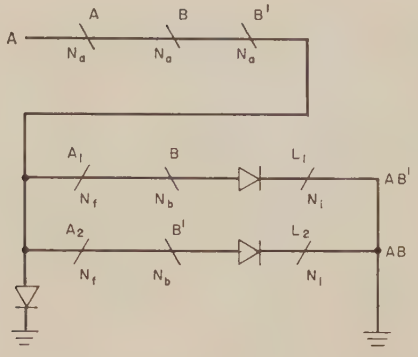



Fig. 6—Illustrating a restriction on multiple-output circuits.

Diodes must be placed in output networks so that no circulating currents can exist. Such circulating currents, whether occurring during the input phase or during the advance phase, could improperly switch cores with windings in their paths. One method of properly placing diodes is to assume one diode with the proper polarity is placed in each path between every pair of nodes. The absence of some of these diodes will permit circulating currents or permit current to flow in the wrong direction. The other diodes are superfluous.

## Economical Logical Design

An economical realization of a given switching function incorporates a minimal number of magnetic cores, windings, diodes, and man-hours spent in logical design. The problem of minimizing magnetic cores is a fascinating topic, and investigating core minimization does shed some light on other problems.

In Fig. 2 the demonstration function is realized using only five cores. If the output windings were all forward or all backward, eight cores would be needed—primed and unprimed inputs for all four variables. The following discussion describes a useful technique for minimization. It is possible to form three of the four combinations of two variables and their complements using only two magnetic cores, in conjunction with other cores representing the other variables. In Fig. 7 cores $x_i$ and $x_j$ are used to deliver product terms involving $x_i'x_j'$, $x_i'x_j$, and $x_ix_j'$. The restriction that no more than one of these two cores may be wound in the forward direction is the reason that $x_ix_j$ cannot be produced. If all four combinations are required by any function, then at least three cores must be used. In the light of these facts, a switching function may be checked to determine which pairs of variables appear with three different combinations of primed and unprimed terms, and which ones appear with all four combinations. This indicates for which pairs of variables two cores may suffice and for which pairs three are needed. It is then necessary to test the various possibilities. One of several different expressions (disjunctive) of the given function may yield a network using fewer cores.

The final step in minimization is to reduce the number of output windings. This reduction is similar to the

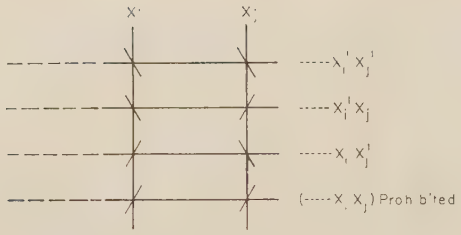



Fig. 7—Producing product terms with two cores.

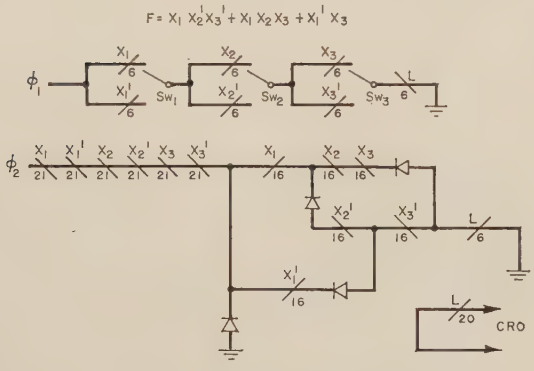



Fig. 8—Typical circuit for producing a function of three variables.

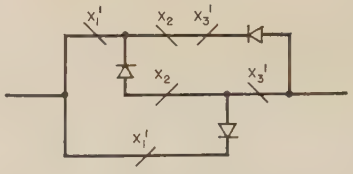


Fig. 9—Alternative output network.

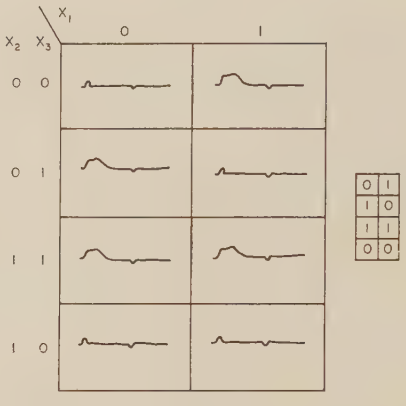


Fig. 10—Output waveforms for core circuit producing function of three variables.

reduction of the number of contact pairs in relay contact networks, with the exception that output windings conduct current in one direction only.

## Experimental Studies

Fig. 8 is the schematic of an experimental circuit which sets a load core, $L$, when the inputs satisfy the equation $F = x_1x_2'x_3' + x_1x_2x_3 + x_1'x_3 = 1$. (Note that this function can be produced with three cores—Fig. 9.) Arnold Engineering Company permalloy tape cores (20 wrap, $\frac{1}{4}$ mil) and Transitron Electronic Corporation T7G diodes were used. The number of turns per winding were calculated according to well-established principles [11]. Negative current pulses of 100 milliamperes amplitude and 2 microseconds duration were applied alternately at the terminals labeled $\phi_1$ and $\phi_2$. During $\phi_1$ one of each pair of cores representing primed and unprimed variables was set, depending upon the positions of the input switches; also, the load core was reset. During $\phi_2$ the output was delivered to the load. The switching of the load core was observed across 20 turns connected to an oscilloscope. Fig. 10 is a series of traces taken from oscilloscope photographs. They represent the voltage across the sensing winding on the load core (shown in the lower right portion of Fig. 8) during $\phi_1$. The traces correspond directly to the Karnaugh map, shown to the right. Notice the small "blips" which appear, whether or not $F = 1$, at the beginning and end of the input pulses of $\phi_1$. These are shuttle voltages (described previously [5]) which cannot be eliminated.

A circuit was constructed to produce the demonstration function of four variables previously discussed. This circuit also performed most satisfactorily. When the circuits were built with different cores, or when the cores were interchanged, no difference in behavior could be observed. This interchangeability is the feature of the new system which makes it so ideal for application to standard packages.

One problem encountered was the slight switching of load cores which occurred when no output signal was supposed to be delivered to the load. The reason for this malfunction is that, for some combinations of inputs, when $F = 0$ none of the cores corresponding to windings in a path may be switching, so no net voltage is generated along the path. This occurs for the top path of the output network of Fig. 8 when $(x_1, x_2, x_3) = (0, 0, 0)$. Therefore a part of the advance current may sneak down this forward path and partially switch the load instead of flowing through the shunt diode. When an extra diode was placed in series with the input winding of the load core, the partial switching was greatly limited. This diode effectively increases the impedance of the load path relative to that of the shunt path.

It was originally assumed that $N_b$ must be greater than $N_f$, to insure a net backward voltage when a core with a forward winding and one with a backward winding in the same path are switching simultaneously. Experimental studies revealed that making $N_b$ equal to $N_f$ did not appreciably increase the noise seen at the output when $F = 0$. Hence, all experimental circuits were constructed with $N_b = N_f = 16$ turns. This significant fact implies that only three types of windings are needed on any core: input windings (6 turns in the experimental circuits), one advance winding (21 turns), and forward-backward windings (16 turns). The only difference

between forward and backward windings is the way they are connected into circuits; *i.e.*, a forward winding becomes a backward winding if its terminals are interchanged. Since only three types of windings are required, the expense of winding the cores is considerably reduced.

Tests were made to see how sensitive the experimental circuits were to changes in the characteristics of the current pulses furnished by the core driver. Circuit behavior was not significantly affected by moderate variation in pulse width, pulse amplitude, and pulse rise time. An increase in pulse width above the necessary minimum did not change the response of the circuits. The test circuits also operated properly for a very large range of pulse amplitude. Increasing $I$ above the specified value merely speeded up the circuit operation; cores switched faster, but the circuits still operated according to logical specifications.

## Standard Packages

The actual packaging of core circuits must be done with the over-all system in mind. After the designer has derived the proper number of turns for the three types of windings, he must decide upon convenient numbers of cores to include in the packages, and how many input windings and forward-backward windings to wind on each core. His object is to be satisfied with as few different types of packages as are consistent with economical use of the components.

The components might be cast in an epoxy type of resin. (Librascope manufactures its "Decision Elements" in that form.) An actual 6-core package might consist of 6 cores and 5 diodes. There would be an advance winding on each core, and the cores would be connected in series, with just the two extremities appearing at surface terminals. The input windings and forward-backward windings would be terminated at the surface. Diodes also would be terminated at the surface of the package, and there might be several sets of shunted terminals to serve as nodes. A particular switching function would be produced by properly interconnecting the terminals on a standard package.

## Comparisons and Conclusions

Among existing systems of core logic, the ones most suitable for standardization are those of Minnick [4], Karnaugh [5], and Andrews [6]. These systems do have their drawbacks. Minnick was one of the first (1953) investigators of magnetic core pulse-switching. The circuits he described generally require several clock pulses per cycle of operation; they cycle all cores once every cycle of operation, deliver unwanted outputs before the desired output is delivered, and use a comparatively large number of components (cores, diodes, windings). Karnaugh's "A-type" circuits remedied these shortcomings. However, the "A-type" circuits demand that all input pulses arrive simultaneously. In devices where input pulses must travel from different physical locations, it is likely that the inputs will not arrive simultaneously. Andrews, with his "shunt-type" circuit, solved this problem by moving the logic function from the input windings to the output windings. In a "shunt-type" circuit inputs may arrive in any order whatsoever. The difficulties with "shunt-type" logic are an unwanted pulse of output current which appears at the beginning of the advance pulse, and a greater number of cores than found in "A-type" circuits.

These final problems are solved in the new type of logic circuits. No noise pulse appears. Furthermore, the number of cores, diodes, and windings which are used in the new system compare favorably with the numbers required in equivalent "A-type" circuits. The primary disadvantage of the new system, however, is the relatively large number of turns which must be wound on the cores.

The salient advantage of the new system is the ease with which circuits can be constructed. Once all cores are wound with the appropriate number of turns in each winding, circuits are built by interconnecting windings according to sketches similar to Fig. 2. The fact that any core may be chosen to represent any variable—*i.e.*, that circuit behavior is not dependent upon which cores are used where—makes the new type of logic an especially good choice for use in a system of standard packages.

The experimental studies of the new system demonstrated the practicability of building combinational switching circuits using advance current drive, with logic performed by forward and backward windings in an output network. It was observed that such circuits may be designed to be independent of moderate changes in driving pulse amplitude, duration, and rise time. Moreover, the behavior is not affected by interchanging components.

## Acknowledgment

The author is grateful to Dr. E. J. McCluskey for his advice and support. The help and suggestions of Prof. S. H. Caldwell, Rhoda Green, Dr. J. E. Mack, M. C. Paul, and S. H. Washburn were invaluable. This work was done with the kind support of the Bell Telephone Laboratories.

## Bibliography

[1] M. K. Haynes, "Magnetic cores as elements of digital computing systems," thesis, University of Illinois, Urbana, Ill.: August, 1950.
[2] R. A. Ramey, "The single-core magnetic amplifier as a computer element," *Trans. AIEE*, vol. 71, pp. 442–446; 1952.
[3] N. B. Saunders, "Magnetic binaries in the logical design of information handling machines," *Proc. Assoc. for Computing Mach.;* May 2–3, 1952.
[4] R. C. Minnick, "The use of magnetic cores as switching devices," thesis, Harvard University, Cambridge, Mass.; April, 1953.
[5] M. Karnaugh, "Pulse-switching circuits using magnetic cores," Proc. IRE, vol. 43, pp. 570–583; May, 1955.
[6] F. T. Andrews, Jr., "Shunt type magnetic core logic circuits," unpublished work.

[7] D. Loev, W. Miehle, J. Paivinen, and J. Wylen, "Magnetic core circuits for digital data-processing systems," PROC. IRE, vol. 44, pp. 154–162; February, 1956.
[8] Minnesota Electronics Corp., Tech. Bull. 114-50153, 116-50153, 117-50153, 111-100153, 112-50553, and Tech. Bull., Librascope Inc., Burbank Div., p. 246; May, 1953.
[9] E. A. Sands, "The behavior of rectangular hysteresis loop magnetic materials under current pulse conditions," PROC. IRE, vol. 40, pp. 1246–1250; October, 1952.
[10] M. Karnaugh, "The map method for synthesis of combinational logic circuits," *Trans. AIEE (Commun. and Electronics)*, vol. 72, pp. 593–599; November, 1953.
[11] F. T. Andrews, Jr. and D. B. James, "Design of magnetic core circuits," unpublished work.
[12] I. L. Auerbach and S. B. Disson, "Magnetic elements in arithmetic and control circuits," *Elec. Eng.*, vol. 74, pp. 766–770 September, 1955.
[13] S. Guterman, R. D. Kodis, and S. Ruhman, "Logical and control functions performed with magnetic cores," PROC. IRE, vol. 43, pp. 291–298; March, 1955.
[14] R. C. Minnick, "Magnetic switching circuits," *J. Appl. Phys.*, vol. 25, pp. 479–485; April, 1954.
[15] S. H. Caldwell, "Switching Circuits and Logical Design," John Wiley and Sons, Inc., New York, N. Y., pp. 363–380, 647–655; 1958.
[16] J. A. Rajchman, "Static magnetic matrix memory and switching circuits," *RCA Rev.*, vol. 13, pp. 183–201; June, 1952.
[17] J. A. Rajchman and H. D. Crane, "Current steering in magnetic circuits," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 21–30; March, 1957.

# The Switching Characteristics of 4-79 Permalloy Cores with Different Anneals*

T. D. ROSSING†, W. M. OVERN‡, AND V. J. KORKOWSKI‡

*Summary*—**The magnetic properties of 4-79 Permalloy cores, which have been annealed at different temperatures, are discussed. Cores annealed at relatively low temperatures are characterized by high coercivity, slower switching, insensitivity to strain, magnetization difficult to rotate, and insensitivity to an applied transverse field. Some cores exhibit a preference to one remanent state over the opposite state.**

DURING the attempted development of a memory with nondestructive readout, investigations have been made of the switching characteristics of 4-79 Moly-Permalloy, ferrites, and vacuum-deposited films. For reasons which will become apparent in this paper, Permalloy tape cores are not too promising in this application. However, some of the data gathered in these investigations sheds new light on the remagnetization process in ferromagnetics.

Several experiments[1] have shown that the wall-motion switching model describes the remagnetization of Permalloy tape cores. For switching times of 10 microseconds or less, the switching time is related to the applied field $H$ by the equation:

$$(H - H_0)t = S.$$

$H_0$ is approximately the same as the coercivity $H_c$, and $S$ is a switching coefficient which depends upon the number of domains of reverse magnetization nucleated and the speed at which the walls of these domains move through the specimen.

The structure of the cores used in this investigation is shown in Fig. 1. Two $\frac{1}{8}$-mil 4-79 Permalloy ribbons, $\frac{1}{8}$ inch wide, are spot welded to stainless-steel straps coated with insulation. They are then wound around the bobbin and spot-welded to a stainless-steel band which constitutes the outer wrap of the core. By means of the attached current leads, current may be passed through the wraps of the core. The magnetic field produced by such a current will be everywhere transverse to the magnetization of the ribbon. These cores were annealed for one hour in a hydrogen atmosphere at various temperatures ranging from 425° to 900°C. The resulting coercivities of these cores are shown in Fig. 2. It is interesting to note that the lower the coercivity of the core, the more sensitive it becomes to strain. Cores with coercivities of 0.6 oersted or more have been completely unwrapped and rewrapped after annealing without any appreciable change in magnetic properties.

The switching coefficient $S$ also depends upon the coercivity of the core. In general, the higher the coercivity of the core, the larger $S$ becomes. Fig. 3 is a plot of $S$ as a function of $H_c$ for cores from various anneals. On the other hand, as the coercivity is lowered, the transverse field effect on the switching behavior is diminished. For faster switching, therefore, using the coincidence of longitudinal and transverse field pulses, the optimum coercivity will be determined by a compromise between these two effects. In most high-speed computer memories, writing is accomplished by the coincidence of two "half-select" pulses which are current pulses of half amplitude. The same compromise must be made, therefore, when writing is accomplished by this coincident-current technique since the switching speed increases as $H_c$ is raised but decreases with increasing $S$.

A most disturbing feature of tape-wound cores of

* Manuscript received by the PGEC, March 27, 1958. This research was supported in part by the USAF.
† Dept. of Phys., St. Olaf College, Northfield, Minn.
‡ Remington Rand Univac, St. Paul, Minn.
[1] N. Meuyuk and J. B. Goodenough, "Magnetic materials for digital computer components. I. Theory of flux reversal in ferromagnetics," *J. Appl. Phys.*, vol. 26, p. 8; January, 1955.

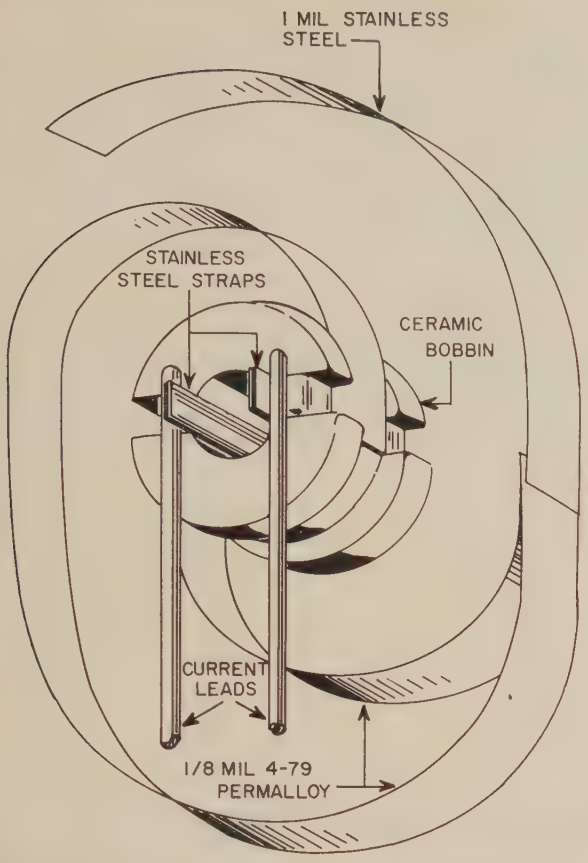



Fig. 1—Core used for transverse field effect measurements.

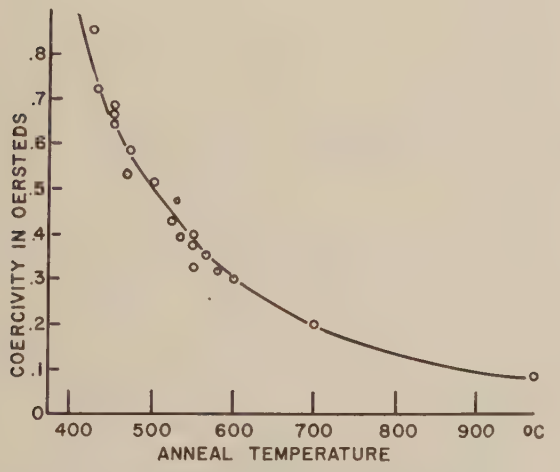



Fig. 2—Coercivity as a function of anneal temperature. All anneals were made for one house in a hydrogen atmosphere.

4-79 Permalloy is their tendency to "creep" into the demagnetized state or into the opposite remanent state when a large number of "half-select" pulses are applied. In many cases, creep can be brought about by applying pulses smaller than the 60-cycle coercivity $H_c$. To minimize magnetic creep in the unselected cores, bidirectional writing pulses have been used. The relative amplitudes of the positive and negative pulses, however, must be carefully controlled.

A block diagram of a dynamic core tester is shown in Fig. 4. "Read" pulses consist of 0.1-$\mu$sec current pulses

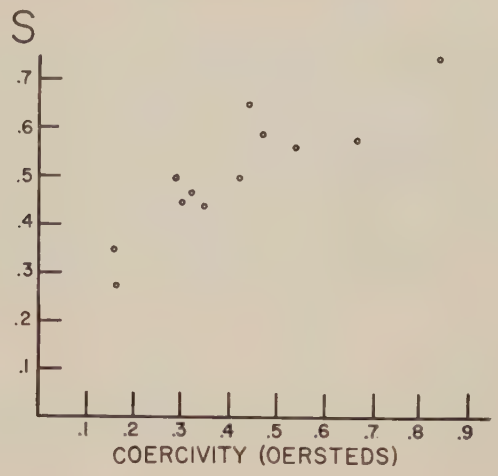



Fig. 3—Switching coefficient $S$ as a function of coercivity.

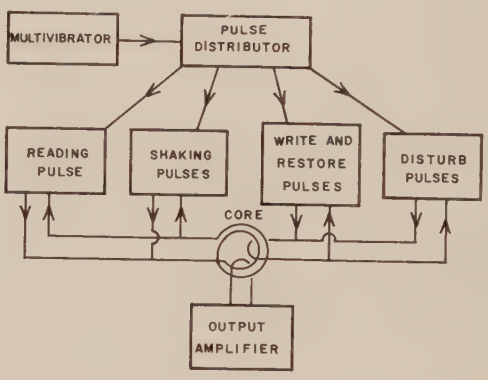



Fig. 4—Block diagram of a dynamic core tester.

applied through the magnetic tape itself to produce a transverse magnetic field. The magnetic state of the core thus may be nondestructively sensed at any point in the test cycle. "Shaking" pulses are bursts of these same 0.1-$\mu$sec pulses applied at the 2-mc rate. These shaking pulses produce the transverse field which switches a core when it coincides with a switch (longitudinal field) pulse. In a typical test program, a restore pulse is followed by a read, a write pulse, another read, and then a large number of disturb pulses alternating in direction. Another read pulse then shows how much change in magnetization (creep) has been brought about by the disturb pulses. This test program is shown in Fig. 5.

Fig. 6 shows the results obtained from one of the most stable cores tested with write and disturb pulses 7 $\mu$sec long. The stable region is bounded by curves showing the maximum amount by which the second half of the bidirectional write pulse may be larger or smaller than the first half without causing creep. Also indicated are the switching fields necessary to switch the magnetic state of the core in 7 $\mu$sec with ($H_s$) and without ($H_n$) a transverse shaking field. Also shown, for purposes of comparison, are the coercivity $H_c$ and $H_{\min}$, the field below which the hysteresis loop completely collapses. It is interesting to note that in this core, writing could be accomplished with a 0.49-oersted switch pulse and a burst of shaking pulses of 800 ma. Coincident-current

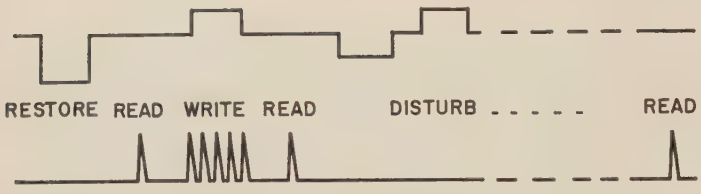



Fig. 5—Typical pulse program used to test for magnetic "creep" with the dynamic core tester.

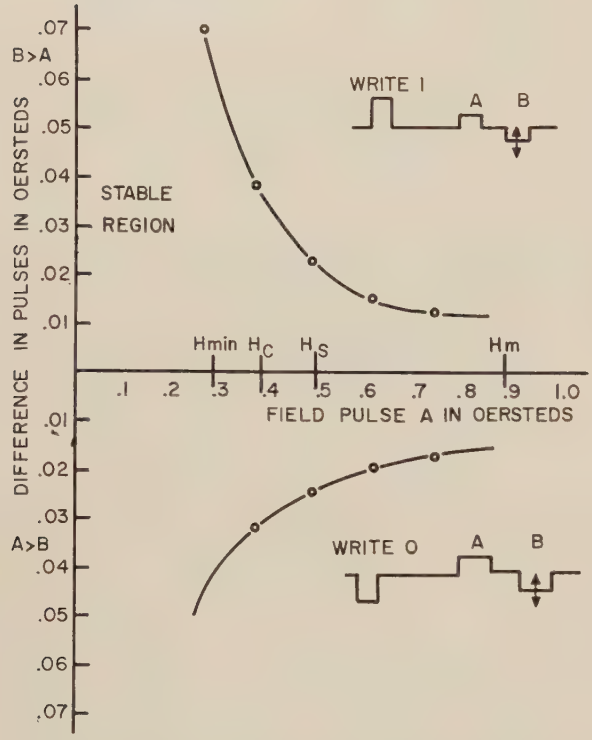



Fig. 6—A stability plot obtained from one of the most stable cores. The two curves indicate the maximum difference in magnetic field pulses which can be tolerated in the "1" and "0" states of remanence.

writing would require half-select ($\frac{1}{2} H_m$) pulses of about 0.49 oersted. To prevent creep, the pulse amplitudes would have to be controlled to within plus or minus 4 per cent.

An average core as regards creep behavior is shown in Fig. 7. In this core pulses would have to be controlled to plus or minus 1 per cent or more. An interesting effect which is shown in Fig. 7 is that there is a preferred state of remanence into which the core will creep more readily than into the other state. Some such preference was demonstrated by nearly half the cores tested, and was independent of the orientation of the cores in the tester. Magnetic fields as large as 200 oersteds did not reverse the preferential direction, but they increased or diminished the amount of preference, depending upon the direction in which they were applied. Meiklejohn and Bean[2] have observed a similar preferential direction

[2] W. H. Meiklejohn and C. P. Bean, "New magnetic anisotropy," *Phys. Rev.*, vol. 102, p. 1413; June, 1956.

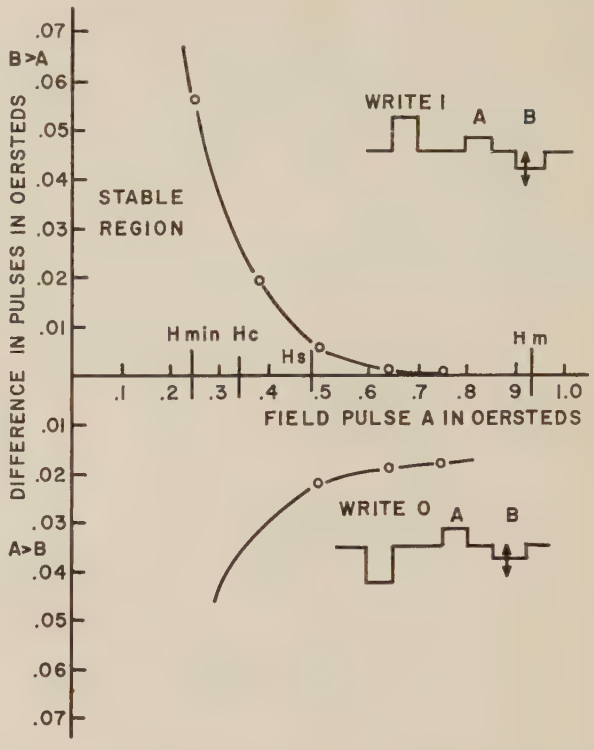



Fig. 7—A stability plot for an "average" core.

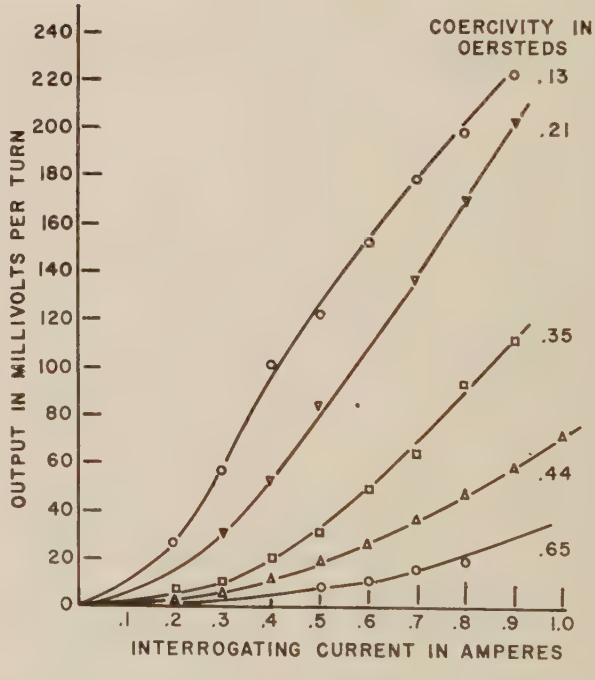



Fig. 8—Outputs obtained when nondestructive readout interrogating pulses are applied to cores of different coercivities. Such outputs are measures of the ease of rotation of the remanent magnetization.

phenomenon which, they explain, is an interaction between an antiferromagnetic and a ferromagnetic material.

Another property of the cores which depends on coercivity is the ease with which the magnetization can be rotated by a transverse field. A convenient measure of this property is the amplitude of the nondestruc-

tive readout signal obtained when a 0.1-$\mu$sec transverse magnetic pulse is applied. Fig. 8 is a plot of readout signals as functions of applied 0.1-$\mu$sec pulses for cores of varying coercivity. Apparently, in cores of low coercivity the magnetization may be rotated reversibly with more ease than in cores of high coercivity. This same effect has been noted in ferrite memory cores.

### Conclusions

Cores of 4-79 Molybdenum Permalloy can be made to have a wide variety of magnetic properties depending upon at what temperature they are annealed. An anneal at about 575°C produces the best cores for memory application.

Cores annealed at relatively low temperatures are characterized by: 1) high coercivity, 2) large value of the switching coefficient $S$, 3) insensitivity to strains, 4) magnetization relatively difficult to rotate, and 5) switching time quite sensitive to an applied transverse field. The properties of cores annealed at high temperatures are the opposite. A slight preference to one magnetic state has been noted in both types of core.

### Acknowledgment

The authors are grateful to Dr. J. A. Randmer and his staff at the South Norwalk Laboratory of Remington Rand Univac for preparing the many cores used in these investigations.

# Formal Analysis and Synthesis of Bilateral Switching Networks*

RAYMOND E. MILLER†

*Summary*—**Formal procedures for the analysis and synthesis of two-terminal combinational bilateral switching networks are presented. A bilateral switching network is one which contains only elements having the same switching transmission characteristic in both directions.**

**Following the definitions for the terminology and notation, where some new terms are introduced, the definitions for a series-parallel network, a bridge element, and a bridge network are given. A condition, called the bridge condition, to test a given transmission function for possible bridge network realizations is presented.**

**A stepwise decomposition procedure is developed which may be used for the analysis and synthesis of the series and parallel parts of the network. The steps are described both with linear graphs and connection matrices. The bridge condition partially formalizes bridge network synthesis. Redundant variables also are considered as an aid to network synthesis. Under certain conditions, the synthesis yields network realizations with the fewest possible number of elements.**

## I. Introduction

THE application of Boolean algebra to the analysis and design of switching networks is quite well known. A Boolean function can be realized by an unlimited number of different switching networks. The network chosen usually depends on some type of economy consideration. One type of economical switching network design is one which uses the least possible number of elements; such a network is called a simplest switching network. The general problem of finding such a simplest switching network design to represent a given Boolean function has not yet been solved. The synthesis of simplest switching networks is complicated by the facts that many essentially different network configurations are possible to represent the function, the simplest network is not necessarily unique, and that it is extremely difficult to show that a network is indeed a simplest one.

This paper considers the problem of designing simplest switching networks for a special class of switching networks. An analysis and a synthesis procedure are presented for two-terminal combinational bilateral switching networks. A bilateral switching network is one which contains only elements having the same switching transmission characteristics in both directions. Relay contact networks are the most common example of bilateral switching networks, whereas electronic logical networks do not normally fall into this class of switching networks. The network synthesis procedure given here is developed to yield simplest switching networks, and in certain cases it is proved to do so.

* Manuscript received by the PGEC, April 1, 1958; revised manuscript received, June 24, 1958. This paper is derived from a dissertation submitted in partial fulfillment of the requirements for the Ph.D. degree in electrical engineering at the University of Illinois, Urbana, Ill.

† IBM Corp., Yorktown Heights, N. Y.

The process is capable of synthesizing bridge networks and nonplanar networks as well as series-parallel networks.

The basic postulates and theorems of Boolean algebra are assumed to be familiar to the reader.[1,2] The formulas of Boolean algebra are to have the *letter variables* $a, b, c, \cdots$. The operation of complementation or negation is denoted by a "$-$," for example, $\bar{b}$; that of conjunction or intersection by juxtaposition, for example, $xy$; and that of alternation or union with a "$\vee$," for example, $c \vee d$.

Letter variables and their negations are called *literals*.[3] A conjunction of literals such that no literal appears twice and no literal and its negation appear is called a *clause*. If a literal and its negation appear in a conjunction of literals, then the conjunction is called a *zero value clause*. A literal itself may be called a clause. An alternation of clauses is called an $\vee$ *polynomial*. A clause itself may be considered to be an $\vee$ polynomial. A simplest equivalent to an $\vee$ polynomial is called a *minimum* $\vee$ *polynomial*. Quine calls this a simplest normal equivalent.

Two functions, $f_1$ and $f_2$, are said to be *matchable* or to have a *matching* if and only if for each clause in $f_1$ the same clause appears in $f_2$, and vice versa.

Given an $\vee$ polynomial $f$ such that

$$f = g_1 g_2 \cdots g_p \vee f',$$

then the right member is called an *assembling* of $f$ if and only if upon expanding $g_1 g_2 \cdots g_p$, using only the distributive and commutative laws, the resulting right member is matchable with the given $f$. A function is called *fully assembled* if and only if all of its clauses are used in the assembling process. It follows in this case that $f' \equiv 0$.

A conjunction of some of the literals appearing in a clause is called a *subclause* of that clause. A clause may be considered to be a subclause of itself. A *conjunction (without simplification) of subclauses is considered to be a clause or a subclause* only if no literals appear more than once in the conjunction. In the following sections, all conjunctions of subclauses are considered to be clauses or subclauses unless specifically stated otherwise. Clauses and subclauses are denoted by capital letters and conjunctions of capital letters, for example, $A$, $ACE$, and $SRE$.

The Boolean transmission functions will usually be denoted by the symbol $f$ or $T$, and are not to be confused with the symbols for a literal or a subclause.

The minimum $\vee$ polynomial is most generally used as the starting function for network synthesis. Various procedures for obtaining a minimum $\vee$ polynomial for a given Boolean function are well known.[3–9]

The switching networks are assumed to be depicted as weighted linear graphs. The weights are Boolean functions attached to the elements of the graph. An *element* consists of a line segment and its endpoints. A *vertex* is an endpoint of an element, and a *linear graph* is a collection of elements such that no two elements have a point in common which is not a vertex. A *subgraph* is a graph containing a subset of the elements of the graph. A vertex and an element are *incident* with each other if the vertex is an endpoint of the element. The vertices of the graph may be classified as either *internal vertices* or *terminal vertices*. The transmission functions represent the transmission between terminal vertices. Switching networks represented by such weighted graphs are known as combinational switching networks.

The switching network may also be represented by a *connection matrix*.[10–12] For a graph with $p$ vertices the connection matrix is a $p \times p$ matrix. The matrix entries are Boolean functions. An entry $c_{ij}$ represents the connection from vertex $i$ to vertex $j$. The $c_{ii}$ entries are all defined to be the Boolean 1. If $c_{ij} = 0$, this means that no direct connection exists between vertices $i$ and $j$. If $c_{ij} = 1$, this means that a short circuit exists between vertices $i$ and $j$. Now if $c_{ij} = 0$, 1, a literal, or an alternation of literals for each $c_{ij}$ of a connection matrix, then the matrix is called a *primitive connection matrix*.

A network with two terminal vertices is called a *two-terminal network*. For convenience, the input vertex is labeled vertex 1, and the output vertex is labeled vertex 2. The weights of elements incident to the input vertex thus appear as entries in the first row and column of the corresponding primitive connection matrix. The weights of elements incident to the output vertex appear as entries in the second row and column.

[1] G. Birkhoff and S. MacLane, "Algebra of Classes" in "A Survey of Modern Algebra," The Macmillan Co., New York, N. Y., ch. 11, pp. 311–325; 1948.

[2] R. Serrell, "Elements of Boolean algebra for the study of information handling systems," PROC. IRE, vol. 41, pp. 1366–1380; October, 1953.

[3] W. V. Quine, "A way to simplify truth functions," *Amer. Math. Month.*, vol. 62, pp. 627–631; November, 1955.

[4] Harvard Univ. Computation Lab. Staff, "Synthesis of Electronic Computing Control Circuits" in "Annals of the Computation Laboratory," Harvard University Press, Cambridge, Mass., no. 27; 1951.

[5] M. Karnaugh, "The map method for synthesis of combinational logic circuits," *Trans. AIEE (Commun. and Electronics)*, vol. 72, pp. 593–599; November, 1953.

[6] E. J. McCluskey, Jr., "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.

[7] D. E. Muller, "Minimization of $\vee$ Polynomial Subject to Subsidiary Conditions," Digital Computer Lab., University of Illinois, Urbana, Ill., Internal Rep. No. 51; October, 1953.

[8] J. P. Roth, "Algebraic Topological Methods for the Synthesis of Switching Circuits in $n$ Variables," Inst. for Advanced Study, Princeton, N. J., Electronic Computer Project, Tech. Rep. No. 56-02; April, 1956.

[9] R. H. Urbano and R. K. Mueller, "A topological method for the determination of the minimal forms of a Boolean function," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 126–132; September, 1956.

[10] B. I. Aranovich, "The use of matrix methods in problems of the structural analysis of relay contact networks," *Avtomatika i Telemekhanika*, vol. 10, no. 6, pp. 437–451; 1949.

[11] F. E. Hohn and R. L. Schissler, "Boolean matrices and the design of combinational relay switching circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 177–202; January, 1955.

[12] W. Semon, "Matrix Theory of Switching Networks," Computation Lab., Thesis, Harvard University, Cambridge, Mass.; 1954.

## II. Series-Parallel and Bridge Network Classifications

The switching networks to be analyzed and synthesized are conveniently classified into two types, the series-parallel, and the bridge networks. The Boolean functions may be directly represented by series-parallel networks, and vice versa. Boolean functions also may be used to represent the transmission in a bridge network; however, the representation of a Boolean function by a bridge network is not direct. The definitions of the networks show that a network is either a series-parallel or a bridge network, but never both, for a specified input vertex and output vertex.

Two equivalent definitions for series-parallel networks follow.[13]

*Definition 1:* A network $N$ is series-parallel with respect to the two terminals $a$ and $b$ if through each element of $N$ there is at least one path from $a$ to $b$ not touching any junction twice, and no two of these paths pass through any element in opposite directions.

*Definition 2:* A network is series-parallel if it is either a series or a parallel connection of two series-parallel networks. A single element is a series-parallel network.

To define a bridge network, a definition is given for a bridge element in a network.

*Definition 3:* An element of a graph representing a switching network is a *bridge element* with respect to the assumed input and output vertices if and only if it belongs to a set of elements having the properties:

Removal of this set from the graph reduces it to a series-parallel network;

When the set of elements is removed from the graph no internal vertex has fewer than two incident elements;

No proper subset of elements of this set when removed from the graph will give a series-parallel network;

No element of the set is incident to either the input or the output vertex.

Now, a bridge may be simply defined.

*Definition 4:* A network is a *bridge network* if and only if it contains at least one bridge element in its graphical representation.

Some examples of bridge networks are now given.

*Example 1:* Removal of element $e$ of Fig. 1 yields a series-parallel network. Element $e$ is a bridge element with respect to the terminal vertices 1 and 2. No other elements are bridge elements.

*Example 2:* The bridge element sets for the network of Fig. 2 are: $\{e\}$, $\{d, f\}$. This example shows that more than one bridge element set may exist for a given graph, and also that the number of elements in each bridge element set need not be a constant for a given network.

Some known facts for switching networks are particularly useful in the synthesis procedures to be developed. These are stated as three theorems.

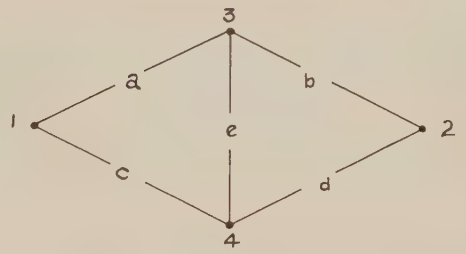



Fig. 1—A simple bridge network.

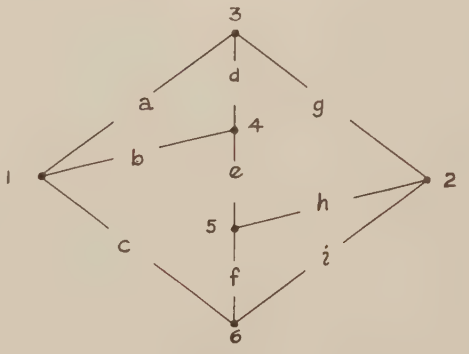



Fig. 2—A bridge network having two bridge element sets.

*Theorem 1:* A network with a minimum number of elements which represents a Boolean transmission contains at least as many elements as the number of different literals appearing in the minimum $\vee$ polynomial representation of the function.

This follows from the definition of a minimum $\vee$ polynomial.[14]

*Theorem 2:* If a switching network with bilateral elements is represented by a connection matrix $A$, cross out the input vertex column and the output vertex row. The determinant formed from this resulting minor yields the transmission function.

This determinant is called an *output determinant* for the network.

The determinant of a Boolean matrix, $A_n$, is defined as the alternation of the $n!$ conjunctions of the entries of $A_n$ in which each row subscript and each column subscript is repeated once and only once in each conjunction.

This theorem is due to Aranovich.[10] and Semon.[12] An example demonstrates this theorem.

*Example 3:* Consider the network graph and connection matrix $C$ shown in Fig. 3.

Now, letting $T_{ij}$ denote the transmission from vertex $i$ to vertex $j$,

$$T_{12} = \det \begin{bmatrix} 0 & 0 & a \\ d & 1 & c \\ b & c & 1 \end{bmatrix} = a \det \begin{bmatrix} d & 1 \\ b & c \end{bmatrix} = acd \vee ab.$$

[13] J. Riordan and C. E. Shannon, "The number of two-terminal series-parallel networks," *J. Math. and Phys.*, vol. 21, pp. 83–93; August, 1942.

[14] R. E. Miller, "Formal analysis and synthesis of bilateral switching networks," Doctoral dissertation, University of Illinois, Urbana, Ill.; 1957. The proof of this theorem and the other results of this paper appear in this dissertation.

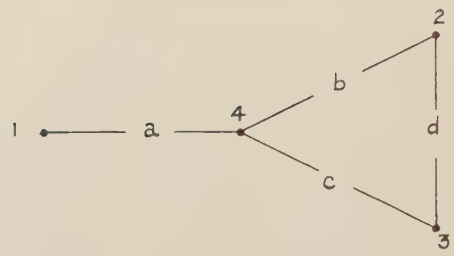



Fig. 3—A network and its associated connection matrix.

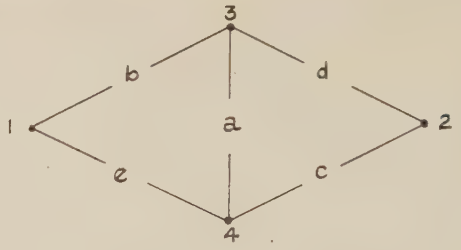



Fig. 4—The simple bridge network for Example 4.

*Theorem 3:* Let $A$ be the connection matrix for a network. Replace each nonzero nonunit entry by the corresponding symbol $a_{pq}$. Form the determinant for the transmisssion function by theorem 2. Then an element represented by $a_{ij}=a_{ji}$ is a bridge element if and only if each of $a_{ij}$ and $a_{ji}$ appears in a separate clause of the expanded $\vee$ polynomial representation (*i.e.*, not in the form $a_{ij}a_{ji}$), where the redundnat clauses are removed.

Semon states this condition as a definition for a bridge element. It is stated here as a theorem and can be shown to follow from Definition 3.

## III. The Bridge Condition

It would be desirable to be able to test a given transmission function in some manner to determine whether the function might be representable as a bridge network with a saving in elements over a series-parallel network. A condition of this sort is given now and it is used in a later section for the synthesis of bridge networks.

Let $A$ represent a subclause of an $\vee$ polynomial, equal to the function $f$. If $A$ is representable as a bridge element, then the function $f$ may be expressed as

$$f = A(BC \vee DE) \vee g,$$

where

1) $A$, $B$, $C$, $D$, and $E$ represent subclauses or literals appearing in $f$ or represent redundant subclauses or literals.
2) $ABC \leq$ the function $f$.
3) $ADE \leq$ the function $f$.
4) $g$ is the remaining part of $f$ with the $ABC$ and $ADE$ related clauses removed.
5) $BD$ is a clause or a subclause of $g$, and CE is a clause or a subclause of $g$, where zero must always be considered as a clause of $g$.

Note that 5) may be restated as

$BD \geqq$ some clause of $g$, and
$CE \geqq$ some clause of $g$.

The symbol $\leq$ means is included in, and $\geq$ means includes.

This condition can be shown to be necessary and sufficient for $A$ to be representable as a bridge element. Some examples will demonstrate the condition.

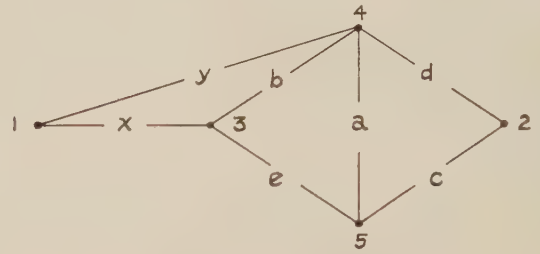



Fig. 5—The bridge network for Example 5.

*Example 4:*

$$f = abc \vee ade \vee bd \vee ce$$
$$= a(bc \vee de) \vee bd \vee ce.$$

Note this function satisfies the bridge condition. The bridge network that realizes this function is shown in Fig. 4.

*Example 5:* Consider the network graph in Fig. 5. For this network

$$f = xbd \vee xabc \vee xce \vee xead \vee yd \vee ybec \vee yac.$$
$$= a(xbc \vee xed) \vee xbd \vee xec \vee yd \vee ybec \vee yac.$$

Since

$(xb)(d) = xbd$ is a clause in $g$, and
$(c)(xe) = xec$ is a clause in $g$,

the element $a$ is seen to satisfy the bridge condition.

## IV. The Decomposition Procedure

A systematic procedure for decomposing a two-terminal switching network is described which provides a method for classifying a given network as either a series-parallel or a bridge network. As described in the next section, the decomposition procedure also is useful for synthesis.

The decomposition procedure is characterized in the form of a graphical analysis, and also as a partitioning and decomposition of the primitive connection matrix. The matrix representation provides a convenient representation for computer use. Let the connection matrix be represented by $C$ and the network graph by $G$. When elements are removed during decomposition, causing a vertex to become isolated, the vertex is also considered removed from the network.

The decomposition procedure is stated in five steps.

*Step (A)—Removal of Series Elements:* If in $G$ a terminal vertex has only one incident element, remove this

element to form a new graph $G'$ with the terminal vertex of $G'$ being that vertex which was incident to the removed element. Reconsider $G'$ as a new $G$. Repeat until no further elements may be removed.

*Step (B)—Removal of Parallel Elements:* Remove all elements through which one and only one path exists between terminal vertices. Let the new graph become a new $G$. The network configuration for any one of the sets of elements in such a path is simply a series network from the input vertex to the output vertex.

*Step (C)—Separation into Series Subnetworks:* In $G$, if all paths from the input vertex to the output vertex pass through a given internal vertex $r$, then break $G$ into two subnetworks $G'$ and $G''$, where $G'$ has input vertex 1 and output vertex $r$ and $G''$ has input vertex $r$ and output vertex 2. Consider each $G'$ and $G''$ as a new $G$, renaming the output vertex of $G'$ as 2, and the input vertex of $G''$ as 1. Repeat this step until no further splitting occurs.

*Step (D)—Separation into Parallel Subnetworks:* If all the internal vertices of $G$ can be separated into two sets such that no element is incident to vertices in both sets, then break $G$ into two subnetworks $G'$ and $G''$, where each subnetwork consists of all the elements and vertices connected to the internal vertices of one of the sets of vertices. Each subnetwork has an input vertex 1 and an output vertex 2 formed at the respective endpoints of the elements incident to the terminal vertices of $G$. Let each $G'$ and $G''$ represent a new $G$. Repeat this step until no further splitting occurs.

*Step (E)—Repetition:* Repeat Steps (A) through (D) on each $G$ until no further reduction can be accomplished in any step.

This procedure reduces the original network to a set of subnetworks or to the null set of elements.

The following properties connected with the decomposition procedure can be shown to hold.

*Property 1:* The decomposition procedure reduces the two-terminal network to a null set of elements if and only if the network is a series-parallel network.

Thus, if the decomposition procedure leaves at least one subnetwork which cannot be reduced further, then the original network is a bridge network.

*Property 2:* A series network is reduced by Step (A) alone to the null set.

*Property 3:* A parallel network is reduced by Step (B) alone to the null set.

*Property 4:* Each subnetwork remaining after the completion of the decomposition procedure is a bridge network.

*Property 5:* In each subnetwork remaining after the completion of the decomposition procedure there exists a "circuit" (*i.e.* a closed path) containing the input vertex and not the output vertex of that subnetwork, and a "circuit" containing the output vertex and not the input vertex such that these circuits have at least one common element.

*Property 6:* The decomposition procedure, when applied to a network, reduces the network to a unique set of subnetworks or to the null set of elements regardless of the order of the decomposition.

The characterization of the network decomposition with the aid of connection matrices follows. Let a primitive connection matrix be

$$C = \begin{bmatrix} 1 & c_{12} & \cdot & \cdot & \cdot & c_{1n} \\ c_{21} & 1 & c_{23} & \cdot & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{n1} & c_{n2} & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

where

1) $n$ is the number of vertices of the network; each vertex is assigned a row and column in $C$,
2) $c_{ij}=c_{ji}$; for $i\neq j$, and
3) Each $c_{ij}$ for $i\neq j$ is 0, a literal, or an alternation of literals.

*Step (A)*: If one and only one $c_{1j}\neq 0$; $j=2, 3, \cdots, n$, say $c_{1p}\neq 0$, then remove row and column 1 from $C$. Permute row and column $p$, of $C$, to the first row and column of the remaining matrix of order $n-1$ to obtain a new connection matrix $C'$. Let $C'$ be a new $C$ and repeat the above process until no further reduction is possible. Repeat similarly on row and column 2.

*Step (B)*: Two separate inspection steps are required to accomplish this step in the matrix representation. They are called (1B) and (2B).

*Case (1B)*: If $c_{12}=c_{21}\neq 0$, replace them with zeros. This removes the single element paths from vertex 1 to vertex 2.

*Case (2B)*: If $c_{1a}\neq 0$, where $a=3, 4, \cdots, n$, and row $a$ has only one other nonzero entry, excluding $c_{a1}$ and $c_{aa}$, call it $c_{ab}\neq 0$, and then inspect row $b$. If row $b$ has only one other nonzero entry, excluding $c_{ba}$ and $c_{bb}$, call it $c_{bd}\neq 0$, and then inspect row $d$. If row $d$ has only one other nonzero entry, excluding $c_{db}$ and $c_{dd}$, call it $c_{de}\neq 0$, where $e\neq a$. Inspect row $e$. Continue this process, never repeating a vertex, until some vertex $r$ is encountered such that $c_{r2}\neq 0$. This ends the sequence. The sequence is $c_{1a}, c_{ab}, c_{bd}, c_{de}, \cdots, c_{r2}$: where in general for the $c_{ij}$'s, no $i$ equals any other $i$ and no $j$ equals any other $j$. If such a sequence is obtainable, remove rows and columns $a, b, d, e, \cdots$, and $r$ from the connection matrix to obtain a new connection matrix $C'$. Let $C'$ be a new $C$ and repeat (2B) until no further reduction is possible.

A partitioning of $C$ may be made as follows if and only if Case (2B) applies. We have

$$C = \left[\begin{array}{c|c|c} C^{11} & C^{12} & C^{13} \\ \hline C^{21} & C^{22} & C^{23} \\ \hline C^{31} & C^{32} & C^{33} \end{array}\right],$$

where rows and columns 3 through $n$ have been permuted in $C$ in such a way that:

1) $C^{11}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, since $c_{12}=0$ from step (1B),

2) $C^{23}=0$ and $C^{32}=0$,

3) All rows of

$$\left[\begin{array}{c|c|c} C^{21} & C^{22} & C^{23}\end{array}\right]$$

have two and only two $c_{ij}\neq 0;\ i\neq j$, and,

4) All columns of

$$\left[\begin{array}{c} C^{12}\\ \hline C^{22}\\ \hline C^{32}\end{array}\right]$$

have two and only two $c_{ij}\neq 0;\ i\neq j$.

Applying Case (2B) gives the new connection matrix $C'$, where

$$C'=\left[\begin{array}{c|c} C^{11} & C^{13}\\ \hline C^{31} & C^{33}\end{array}\right].$$

Let $C'$ be a new $C$ and repeat the partitioning process as often as possible.

*Step* ($C$): Step (C) is applicable to the primitive connection matrix if and only if there exists a partitioning —after suitable permutations of the rows and columns, if necessary—such that

$$C=\left[\begin{array}{c|c|c} C^{11} & C^{12} & C^{13}\\ \hline C^{21} & C^{22} & C^{23}\\ \hline C^{31} & C^{32} & C^{33}\end{array}\right],$$

where

1) $\left[\begin{array}{c|c|c} C^{21} & C^{22} & C^{23}\end{array}\right]=[c_{r1}\cdots c_{rn}]$

$=$ the row for vertex $r$,

2) $\left[\begin{array}{c} C^{12}\\ \hline C^{22}\\ \hline C^{32}\end{array}\right]=\left[\begin{array}{c|c|c} C^{21} & C^{22} & C^{23}\end{array}\right]^T$

$=$ the column for vertex $r$.

3) $C^{13}=0$ and $C^{31}=0$,

4) $\left[\begin{array}{c|c|c} C^{11} & C^{12} & C^{13}\end{array}\right]$

contains the input vertex row, and

$$\left[\begin{array}{c|c|c} C^{31} & C^{32} & C^{33}\end{array}\right]$$

contains the output vertex row.

The two subnetworks $G'$ and $G''$ are then defined by the new primitive connection matrices $C'$ and $C''$, respectively.

$$C'=\left[\begin{array}{c|c} C^{11} & C^{12}\\ \hline C^{21} & C^{22}\end{array}\right] \text{ and } C''=\left[\begin{array}{c|c} C^{22} & C^{23}\\ \hline C^{32} & C^{33}\end{array}\right]$$

The

$$\left[\begin{array}{c} C^{12}\\ \hline C^{22}\end{array}\right]$$

column and the

$$\left[\begin{array}{c|c} C^{21} & C^{22}\end{array}\right]$$

row then are permuted to the second column and row of $C'$, which is the normal position designated for the output vertex in this analysis. This gives the conventional connection matrix $C$ for the reduced network represented by $C'$. The output vertex $C''$ (*i.e.*, the vertex 2 in $C$) is similarly transposed to the second row and column of $C''$ to yield a new $C$ for further analysis.

*Step* ($D$): A partitioning of $C$, as shown below, exists if and only if Step (D) applies.

$$C=\begin{array}{c} \\ 1\\ 2\\ a\\ \\ b\end{array}\begin{array}{c} \begin{array}{ccc} = & 1\ 2 & a \end{array} \quad b\\ \left[\begin{array}{c|c|c} C^{11} & C^{12} & C^{13}\\ \hline & & \\ \hline C^{21} & C^{22} & C^{23}\\ \hline & & \\ \hline C^{31} & C^{32} & C^{33}\end{array}\right]\end{array}$$

where

1) $C^{11}=\begin{bmatrix}1 & c_{12}\\ c_{21} & 1\end{bmatrix}$,

2) The rows and columns denoted by $a$ are from one set of vertices of Step (D), and the rows and columns denoted by $b$ are from the other set of vertices of Step (D),

3) $C^{23}=0$ and $C^{32}=0$.

The primitive connection matrices for $G'$ and $G''$, designated by $C'$ and $C''$ respectively, are

$$C'=\left[\begin{array}{cc|c} 1 & 0 & \\ 0 & 1 & C^{12}\\ \hline \multicolumn{2}{c|}{C^{21}} & C^{22}\end{array}\right] \text{ and } C''=\left[\begin{array}{cc|c} 1 & 0 & \\ 0 & 1 & C^{13}\\ \hline \multicolumn{2}{c|}{C^{31}} & C^{33}\end{array}\right].$$

The omitted subnetwork designated by $C^{11}$ in 1) above need not be considered since it reduces to the null set of elements by Step (B) if $c_{12}$ and $c_{21}$ are not already zero. Let $C'$ and $C''$ each represent a new $C$ for further decomposition.

*Step (E)*: Repeat Steps (A) through (D) on each $C$ until no further reduction can be accomplished in any step.

Some observations concerning the decomposition procedure and its application are in order.

1) A subnetwork with three or fewer vertices will always reduce to the null set of elements since a bridge network requires four or more vertices.

2) Step (B) of the decomposition procedure is a special case of an iteration of Steps (D) and (A). In theory it may be removed from the procedure. It is useful, however, because it reduces the number of iterations required to reduce a network, in many cases. It also allows network classification by Property 3.

3) Steps (A), (C), and (D) are all required steps for a complete decomposition procedure of this type. No one of them may be removed from the procedure without destroying the decomposition process.

4) The location of the zero entries in the connection matrix governs the partitionings which are possible in the matrix procedure.

## V. Two-Terminal Switching Network Synthesis

The starting point for the synthesis of a two-terminal switching network is considered to be the minimum $\vee$ polynomial representation for the transmission function, where "don't care" conditions may be included in the determination of the minimum $\vee$ polynomial.

The procedure to be developed attempts to synthesize a network which attains the lower bound stated in Theorem 1, or when this is not possible, to form a network with the smallest possible number of additional elements. The obviously uneconomical network representations thus are excluded from consideration in this procedure.

In Part A of this section, the steps of the decomposition procedure are restated as conditions for decomposing the transmission function into subfunctions with fewer literals. These steps also specify the network synthesis using connection matrices. This decomposition synthesizes the parts of the function which certainly are represented most simply as series or parallel subnetworks.

In Part B the bridge condition is applied to the functions which are not decomposable to investigate their synthesis into simplest bridge networks.

The use of redundant elements for both series-parallel elements and bridge elements is discussed in Part C.

### A. Series-Parallel Network Elements

*Step (A)*: The network may be synthesized with a single series element incident to a terminal vertex if and only if the minimum $\vee$ polynomial, $f$, may be fully assembled to give

$$f = xf'$$

where $x$ is some literal that appears in each clause of $f$. The output connection matrix for $f$ is

$$C = \begin{bmatrix} 1 & f \\ f & 1 \end{bmatrix}.$$

The partially expanded connection matrix for $f$ may then be represented as

$$C = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & f' \\ x & f' & 1 \end{bmatrix} \quad \text{or} \quad C = \begin{bmatrix} 1 & 0 & f' \\ 0 & 1 & x \\ f' & x & 1 \end{bmatrix},$$

which corresponds to the literal $x$ being incident to the input vertex or the output vertex respectively.

Further synthesis introduces more vertices into the network (*i.e.*, more rows and columns into the connection matrix). Then, in the matrix, zeros are inserted as the new entries in the row and column corresponding to the vertex to which $x$ is incident. Only the function $f'$ need be considered for further synthesis. This means that only the submatrix

$$\begin{bmatrix} 1 & f' \\ f' & 1 \end{bmatrix}$$

requires further expansion. Let $f'$ be a new $f$ and repeat this step as long as it applies.

*Step (B)*: If in $f$ there exists an alternation of clauses, $B_1$, for which all the literals in each of these clauses appear in no other clause of $f$, then each of these clauses should be represented as a separate series path from the input vertex to the output vertex for a simplest network representation. Here, $f = B_1 \vee f'$; where $f'$ represents the alternation of the clauses of the minimum $\vee$ polynomial for $f$ not in $B_1$.

More paths in the final network realization may also be removable by Step (B) of the analysis which are not evident at this stage of synthesis. The output connection matrix for $f$ is

$$C = \begin{bmatrix} 1 & f \\ f & 1 \end{bmatrix} = \begin{bmatrix} 1 & B_1 \vee f' \\ B_1 \vee f' & 1 \end{bmatrix}.$$

The $B_1$ part of the network may be totally synthesized now, and could easily be expressed in primitive matrix form, and $f'$ may be treated as a new $f$ for further decomposition.

*Step (C)*: A necessary and sufficient condition for Step (C) to be usable for synthesis is that some representation of the transmission function be factorable into a conjunction of two $\vee$ polynomials $f'$ and $f''$. This gives

$$f = (f')(f'')$$

It should be noted that the minimum $\vee$ polynomial form for $f$ may not obviously be factorable, and some form for $f$ using redundant literals or clauses may be required to obtain a factoring. The method to obtain a factoring is not always evident immediately. Some formal method would be desirable.

For $f=f'f''$, the connection matrix

$$C = \begin{bmatrix} 1 & f \\ f & 1 \end{bmatrix} = \begin{bmatrix} 1 & f'f'' \\ f'f'' & 1 \end{bmatrix}$$

may be expanded as

$$C = \begin{bmatrix} 1 & 0 & f' \\ 0 & 1 & f'' \\ f' & f'' & 1 \end{bmatrix}.$$

Now $f'$ and $f''$ may each be considered as a new $f$ with the submatrices

$$C' = \begin{bmatrix} 1 & f' \\ f' & 1 \end{bmatrix} \quad \text{and} \quad C'' = \begin{bmatrix} 1 & f'' \\ f'' & 1 \end{bmatrix}.$$

Step (C) should be repeated as often as possible. In general, for a factoring of $f$ as a conjunction of $k$ factors,

$$f = (g_1)(g_2) \cdots (g_k),$$

and by permuting vertex 2 to the last row and column, the following matrix structure is obtained by repeated expansion:

$$C = \begin{bmatrix} 1 & g_1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ g_1 & 1 & g_2 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & g_2 & 1 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & 0 & \cdot & \cdot & \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot & & 1 & g_k \\ 0 & \cdot & \cdot & \cdot & 0 & & g_k & 1 \end{bmatrix}.$$

Each $g_i$ may be considered as a new $f$ for further synthesis.

The resulting network from Step (C) is not generally a minimum network representation unless the factoring is a fully assembled representation of the minimum $\vee$ polynomial.

*Step (D)*: For $f$ in the minimum $\vee$ polynomial representation, if the clauses may be split into two sets of clauses which have no literals in common Step (D) applies. The function may be written as

$$f = f' \vee f''$$

where $f'$ and $f''$ are the alternations of clauses in the first and second sets respectively. The $f'$ and $f''$ may each be considered as a new $f$ and Step (D) may be repeated until it no longer applies. Assume that the original $f$ for this step is thus split into $m$ separate functions—call them $g_1, g_2, \cdots, g_m$—such that

$$f = g_1 \vee g_2 \vee \cdots \vee g_m.$$

Each $g_j$ may be considered as a new $f$ and further synthesis may be carried out to obtain a primitive connection matrix for each $g_j$. Let these be called $C_j$ and partition each $C_j$; $j=1, 2, \cdots, m$, thus,

$$C_j = \left[\begin{array}{c|c} A_{11}{}^j & A_{12}{}^j \\ \hline A_{21}{}^j & A_{22}{}^j \end{array}\right]$$

where

$$A_{11}{}^j = \begin{bmatrix} 1 & a_{12}{}^j \\ a_{21}{}^j & 1 \end{bmatrix}, \quad A_{12}{}^j = [A_{21}{}^j]^T = \begin{bmatrix} a_{13}{}^j \cdots a_{1n_j}{}^j \\ a_{23}{}^j \cdots a_{2n_j}{}^j \end{bmatrix},$$

and $A_{22}{}^j$ is the remaining submatrix.

The primitive connection matrix representing the network for $f$ then is

$$C = \left[\begin{array}{c|c|c|c} \bigvee_{j=1}^{m} A_{11}{}^j & A_{12}{}^{j_1} & \cdots & A_{12}{}^{j_m} \\ \hline A_{21}{}^{j_1} & A_{22}{}^{j_1} & 0 & 0 \\ \hline \begin{matrix}\cdot\\ \cdot\\ \cdot\end{matrix} & 0 & \begin{matrix}\cdot & \\ & \cdot\end{matrix} & 0 \\ \hline A_{21}{}^{j_m} & 0 & 0 & A_{22}{}^{j_m} \end{array}\right]$$

where

$$\bigvee_{j=1}^{m} A_{11}{}^j$$

is the alternation of the $A_{11}{}^j$'s and $j_1$ to $j_m$ is some arrangement of the indexes 1, 2, $\cdots$, $m$, picked at the discretion of the designer.

*Step (E)*: Repeat Steps (A) through (D) until no further simplification can be accomplished by any step of the procedure.

*Theorem 4:* If the minimum $\vee$ polynomial representation, and fully assembled factors are used in all the steps of repeated application (A), (B), (C), and (D), and if the connection matrix reduces to a primitive connection matrix, then the network representation is a simplest network representation, in series-parallel form.

This theorem follows from Theorem 1 and the structure of decomposition.

The following two theorems show that the minimum $\vee$ polynomial representation is invariant under certain decompositions.

*Theorem 5:* Given a minimum $\vee$ polynomial, $M$, then the alternation of any subset of clauses of $M$ is also a minimum $\vee$ polynomial.

*Theorem 6:* For a given minimum $\vee$ polynomial, $M$, if $M$ can be fully assembled into a conjunction of $\vee$ polynomials, then each of these $\vee$ polynomials is also a minimum $\vee$ polynomial.

The following examples demonstrate the use of the decomposition procedure for synthesis.

*Example 6:* Let $T$ be the transmission function to be synthesized where

$$T = abefpqrs \vee abghpqrs \vee cdefpqrs \vee cdghpqrs \vee \bar{a}lpqr \vee mnpqr.$$

$T$ is in a minimum $\vee$-polynomial. Applying Step (A) successively to $p$, $q$, and $r$ gives the matrix

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 0 & p & 0 & 0 \\ 0 & 1 & 0 & 0 & T' \\ p & 0 & 1 & q & 0 \\ 0 & 0 & q & 1 & r \\ 0 & T' & 0 & r & 1 \end{bmatrix} \end{array}$$

where the row and column numbers indicate the numbers assigned to the vertices, and $T' = abefs \vee abghs \vee cdefs \vee cdghs \vee \bar{a}l \vee mn$. Applying Step (B) to $T'$ gives

$$T' = (\bar{a}l \vee mn) \vee (abefs \vee abghs \vee cdefs \vee cdghs) = B_1 \vee T''.$$

The connection matrix for $T'$ thus may be expanded as

$$C' = \begin{array}{c} \\ 5 \\ 2 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccc} 5 & 2 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & T'' & \bar{a} & m \\ T'' & 1 & l & n \\ \bar{a} & l & 1 & 0 \\ m & n & 0 & 1 \end{bmatrix} \end{array}.$$

Step (A) may be applied then to $T''$ followed by Step (C), giving

$$T'' = s(ab \vee cd)(ef \vee gh).$$

A final connection matrix which realizes the network in a simplest form is thus

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{bmatrix} 1 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & l & n & s & 0 \\ p & 0 & 1 & q & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 1 & r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 1 & \bar{a} & m & 0 & ab \vee cd \\ 0 & l & 0 & 0 & \bar{a} & 1 & 0 & 0 & 0 \\ 0 & n & 0 & 0 & m & 0 & 1 & 0 & 0 \\ 0 & s & 0 & 0 & 0 & 0 & 0 & 1 & ef \vee gh \\ 0 & 0 & 0 & 0 & ab \vee cd & 0 & 0 & ef \vee gh & 1 \end{bmatrix} \end{array}.$$

The graph for this network is shown in Fig. 6.

*Example 7:* Let $f = a\bar{b}c \vee \bar{a}bc$. Applying Step (A) gives $f = c(a\bar{b} \vee \bar{a}b)$, and the connection matrix

$$C = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & a\bar{b} \vee \bar{a}b \\ c & a\bar{b} \vee \bar{a}b & 1 \end{bmatrix}.$$

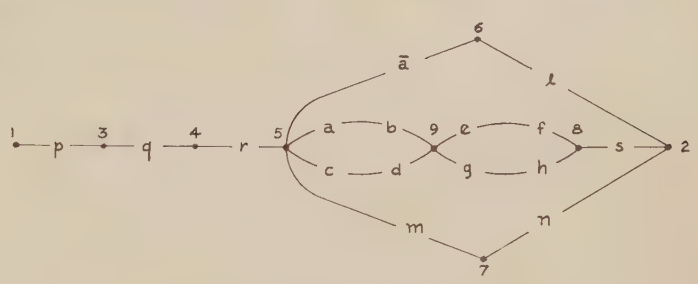



Fig. 6—The network for Example 6.

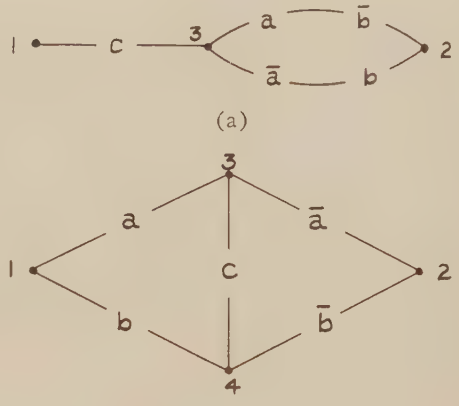



Fig. 7—The network graphs for Example 7.

The corresponding network graph is shown in Fig. 7(a). This is a simplest network realization in series-parallel form. The bridge network seen in Fig. 7(b) is also a simplest network realization for the function.

In general, the bridge condition should be applied at each step of synthesis. It is desirable, however, to obtain the most simple subfunctions possible with the decomposition process before applying the more complex bridge network synthesis process. When this is done, series-parallel network realizations are favored unless simpler bridge network realizations are possible.

### *B. Bridge Network Subfunctions*

Consider network synthesis using a subcondition of the bridge condition given in Section III, where the expression $f = A(BC \vee DE) \vee g$ is an assembling of the minimum $\vee$ polynomial, and $BD$ and $CE$ are clauses in $g$. Let $f = f_b \vee g_1$, where $f_b$ matches $A(BC \vee DE) \vee BD \vee CE$ and $g$ matches $g_1 \vee BD \vee CE$.

If the literals appearing in $A$, $B$, $C$, $D$, and $E$ do not appear in $g_1$, then $f_b$ may be synthesized into a simplest network realization, where the connection matrix for $f$ is

$$C_b = \begin{bmatrix} 1 & g_1 & B & E \\ g_1 & 1 & D & C \\ B & D & 1 & A \\ E & C & A & 1 \end{bmatrix}$$

and the network graph is shown in Fig. 8. An inter-

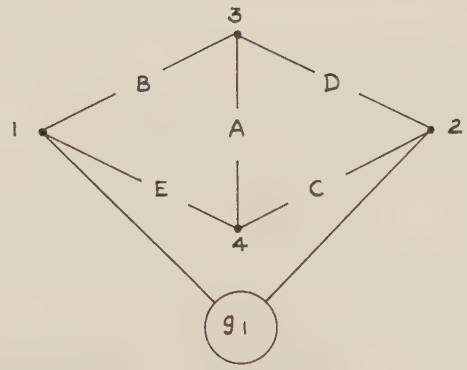



Fig. 8—A partially synthesized bridge network realization.

change of rows and columns 3 and 4, respectively, gives an equivalent network.

Now, take the case that some literals appearing in the subclauses $A$, $B$, $C$, $D$, and $E$ also appear in $g_1$, $g_1 \neq 0$, and also that $f$ is a final subfunction of the decomposition procedure. In this case, $C_b$ represents a partially synthesized network. Further assembling of the clauses of $g_1$ may allow further synthesis. Three such assemblings are illustrated.

1): If $g_1$ contains a clause with the subclause $A$, and one of the subclauses $B$, $C$, $D$, or $E$, then attempt to factor $f$ as

$$f = f_b \vee A(FC_1 \vee D_1E_1) \vee FD_1 \vee C_1E_1 \vee g_2$$

where

$C_1$, $D_1$, and $E_1$ are some three of the subclauses $B$, $C$, $D$, and $E$,
$C_1E_1$ is either $BD$ or $CE$,
$g_2$ is the alternation of the remaining clauses, and
the function $AFC_1 \vee FD_1 \vee g_2$ is matchable with $g_1$.

Under these conditions, the subclause element not represented as $C_1$, $D_1$, or $E_1$ has the element $F$ placed in parallel with it to further synthesize the network. For example, let $B = B_1$, $C = C_1$, $D = D_1$, and $E = E_1$. Then

$$f = A[(B \vee F)C \vee DE] \vee (B \vee F)D \vee CE \vee g_2.$$

The connection matrix and network graph are shown in Fig. 9(a).

2): Suppose $g_1$ contains clauses of the form $HBC$ and $HDE$. Then $f$ may be assembled as

$$\begin{aligned} f &= f_b \vee H(BC \vee DE) \vee g_3 \\ &= (A \vee H)(BC \vee DE) \vee BD \vee CE \vee g_3. \end{aligned}$$

The connection matrix and network graph are illustrated in Fig. 9(b). Element $H$ is parallel with element $A$.

3): Assume that $f$ can be assembled as

$$\begin{aligned} f = {} & A(BC \vee DE) \vee BD \vee CE \vee K(LC \vee ME) \\ & \vee LM \vee AK(BM \vee DL) \vee g_4, \end{aligned}$$

where, if $AK$ has the value zero, the factoring $AK(BM \vee DL)$ is omitted. Such a function may be represented by the connection matrix and network graph in Fig. 9(c).

Application of these three steps with the general bridge condition allows synthesis of "extended simple bridge networks" of the general form shown in Fig. 10. The lines between vertices indicate some functional interconnection.

*Example 8:* A bridge inside a bridge. Let

$$\begin{aligned} f = {} & ac \vee bd \vee aduv \vee advx \vee aduxy \vee advwy \\ & \vee bcuw \vee bcvx \vee bcuxy \vee bcvwy \\ = {} & Q(ad \vee bc) \vee ac \vee bd, \end{aligned}$$

where

$$Q = uw \vee vx \vee uxy \vee vwy.$$

The partially expanded connection matrix may then be written as

$$C_b = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & a & b \\ 0 & 1 & c & d \\ a & c & 1 & Q \\ b & d & Q & 1 \end{bmatrix} \end{array}.$$

The function $Q$ also satisfies the bridge condition, where

$$Q = y(ux \vee vw) \vee uw \vee vx.$$

An expanded connection matrix for $Q$ is

$$C_a = \begin{array}{c} \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & u & v \\ 0 & 1 & w & x \\ u & w & 1 & y \\ v & x & y & 1 \end{bmatrix} \end{array}.$$

The primitive connection matrix for $f$ then becomes

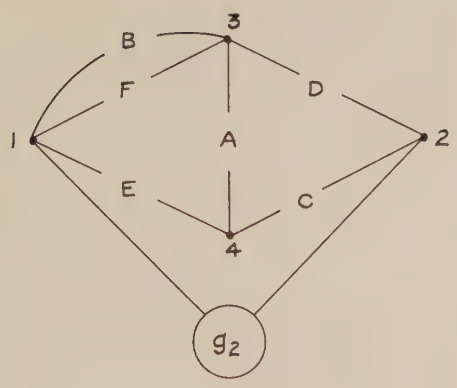
$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & a & b & 0 & 0 \\ 0 & 1 & c & d & 0 & 0 \\ a & c & 1 & 0 & u & v \\ b & d & 0 & 1 & w & x \\ 0 & 0 & u & w & 1 & y \\ 0 & 0 & v & x & y & 1 \end{bmatrix} \end{array}.$$

The corresponding network graph is shown in Fig. 11.

Other extensions of the simple bridge result from factorings similar to those in 1), 2), and 3).

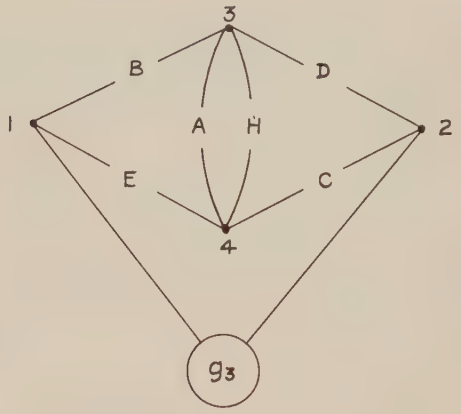
A combination of subnetworks, which are formed by applying the bridge condition synthesis to more than one part of a function, is useful in forming more complex bridge networks. For this, the terms "vertex equality" and "element equality" are defined.

*Definition 5:* A *vertex equality* exists between two vertices in a network if the two vertices may be combined into a single vertex without altering the transmission function of the network.



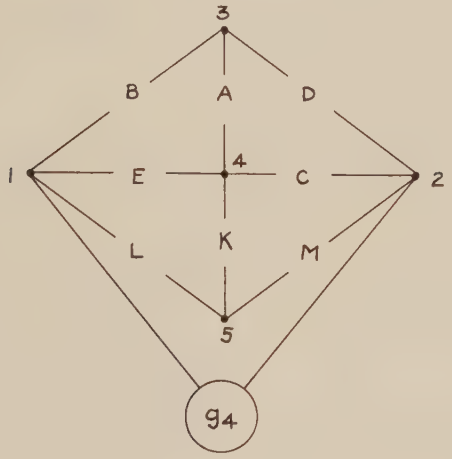

$$C_b = \begin{bmatrix} 1 & g_2 & B \vee F & E \\ g_2 & 1 & D & C \\ B \vee F & D & 1 & A \\ E & C & A & 1 \end{bmatrix}$$

(a)




$$C_b = \begin{bmatrix} 1 & g_3 & B & E \\ g_3 & 1 & D & C \\ B & D & 1 & A \vee H \\ E & C & A \vee H & 1 \end{bmatrix}$$

(b)




$$C_b = \begin{bmatrix} 1 & g_4 & B & E & L \\ g_4 & 1 & D & C & M \\ B & D & 1 & A & O \\ E & C & A & 1 & K \\ L & M & O & K & 1 \end{bmatrix}$$

(c)

Fig. 9—The bridge networks formed by some additional synthesis steps.

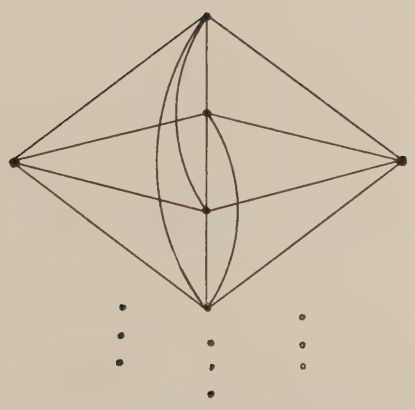


Fig. 10—The extended simple bridge network.

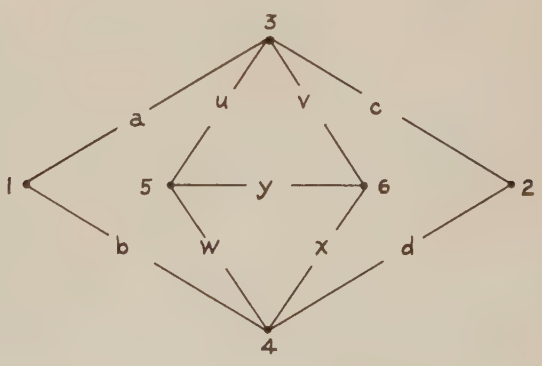



Fig. 11—A "bridge inside a bridge" network.

*Definition 6:* An *element equality* exists between two elements of common weight in a network if the elements may be combined into a single element of the same weight in a network without altering the transmission function of the network.

An example now demonstrates how these equality relations may be used to aid in the network synthesis.

*Example 9:* Consider the transmission function:

$$f = ABL \vee ABCK \vee ADEK \vee BDK \vee BDCL \vee CEK \vee EL.$$

There are two possible ways to apply the bridge condition to $f$. The first is

$$f = f_{b_1} \vee g_1 = A(BCK \vee DEK) \vee BDK \vee CEK \vee ABL \vee BDCL \vee EL,$$

where

$$f_{b_1} = A(BCK \vee DEK) \vee BDK \vee CEK,$$

and

$$g_1 = ABL \vee BDCL \vee EL,$$

which gives the subnetwork for $f_{b_1}$ shown in Fig. 12(a). The second is

$$f = f_{b_2} \vee g_2 = C(ABK \vee BDL) \vee ABL \vee BDK \vee CEK \vee EL \vee ADEK,$$

where

$$f_{b_2} = C(ABK \vee BDL) \vee ABL \vee BDK,$$

and

$$g_2 = CEK \vee EL \vee ADEK,$$

which gives the subnetwork for $f_{b_2}$ shown in Fig. 12(b).

Some interconnection of these two subnetworks to represent $f$ may be possible. The proposed method is to try some interconnection and then test its transmission function by means of the output determinant described in Theorem 2, Section II. For this example the interconnection shown in Fig. 12(c) produces a simplest network realization. The vertex equalities assumed for this interconnection are $3=3'$, $4=4'$, and $5=5'$, and element equalities are assumed for each pair of elements with like weight.

When applying the general bridge condition to network synthesis the $BD$ and $CE$ clauses in the bridge condition may be proper subclauses of some clauses in the minimum $\vee$ polynomial. As stated in the following theorem, this condition does not lead to a necessarily simplest network. Other assembling of the function may give better results.

*Theorem 7:* If the $BD$ and $CE$ clauses in the bridge condition, $A(BC \vee DE) \vee BD \vee CE$, are actually proper subclauses in the minimum $\vee$ polynomial representation for $f$, and if $ABD \vee ACE \vee f \neq f$, then $A$ is not representable as a single bridge element with this method of assembling.

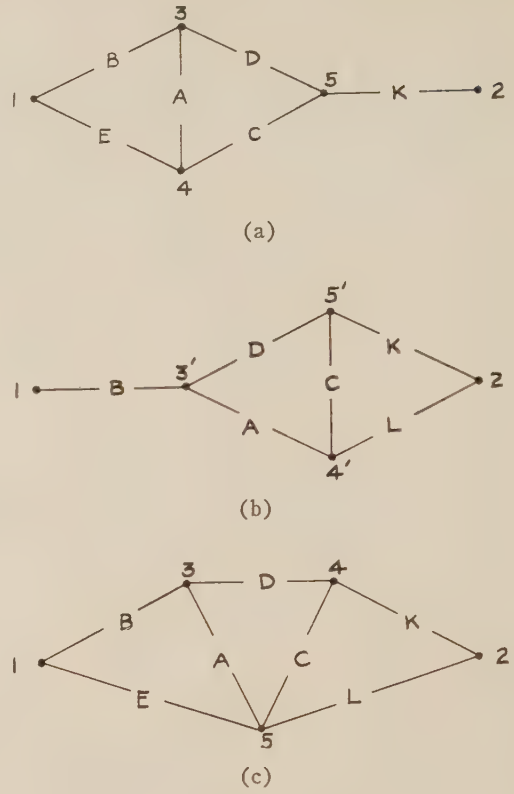



Fig. 12—A network interconnection.

*C. Additional Element Requirements; Redundant Clauses and Literals*

This part considers modification of the minimum $\vee$ polynomial, or its subfunctions, which aid in the synthesis of a simplified network realization of the function. Any such modification of the function must leave the value of the function invariant.

The modifications can be classified into the following four types.

*Type 1:* A zero value clause is joined to the function, where the literals in this clause appear in the functional representation being modified.

*Type 2:* A clause, $B$, which is contained in the function (*i.e.*, $B \leq f$) is joined to the function.

*Type 3:* A clause in $f$ is replaced by some other clause such that the functional value of $f$ remains unchanged.

*Type 4:* The minimum $\vee$ polynomial representation is replaced with some other minimum $\vee$ polynomial representation of the function.

Although Type 2 includes the Type 1 modification, the Type 1 is listed separately since it is quite useful in practice.

The following examples demonstrate the use of modifications on the functional forms to aid network synthesis.

*Example 10:* Let

$$f = ab \vee ad \vee cd \vee bce.$$

$$\text{Type 2} \quad = ab \vee ade \vee ad \vee cd \vee bde$$

$$= e(bc \vee ad) \vee ab \vee cd \vee ad = f_b \vee ad.$$

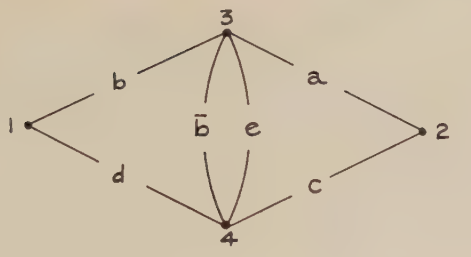



Fig. 13—A bridge network resulting from using redundancies.

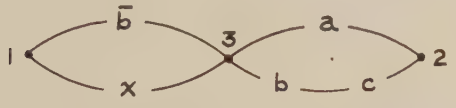



Fig. 14—A series-parallel network formed by using redundancies.

Types 1 and 3 $= e(bc \vee ad) \vee \bar{b}(bc \vee ad) \vee ab \vee cd$

$= (\bar{b} \vee e)(bc \vee ad) \vee ab \vee cd.$

The resulting network graph is shown in Fig. 13.

*Example 11:* Let

$$f = ax \vee bcx \vee a\bar{b} = x(a \vee bc) \vee a\bar{b}.$$

Type 1 $= x(a \vee bc) \vee \bar{b}(a \vee bc) = (\bar{b} \vee x)(a \vee bc).$

The resulting network graph, using Step (C) of the decomposition procedure, is shown in Fig. 14.

These examples illustrate that redundant clauses and literals are useful in network synthesis.

There usually are many ways to modify a given transmission function. One should attempt first to modify the function by using only literals which already appear in the minimum $\vee$ polynomial representation, since such changes may lead to a network synthesis which gives a simplest network as described in Theorem 1, Section II. For such redundancies, Step (C) of the decomposition procedure, and the bridge condition may lead to further synthesis.

The form of the bridge condition indicates the redundancies which might apply to satisfy the bridge condition. Three possible forms are given now.

*Form 1:* The function contains clauses of the type

$$f = A(BC \vee -) \vee BHx \vee CK\bar{x}.$$

The bridge condition factoring may be completed by joining the zero value clause $AHxK\bar{x}$ to $f$, giving

$$f = A(BC \vee HxK\bar{x}) \vee BHx \vee CK\bar{x},$$

which results in the network graph shown in Fig. 15(a).

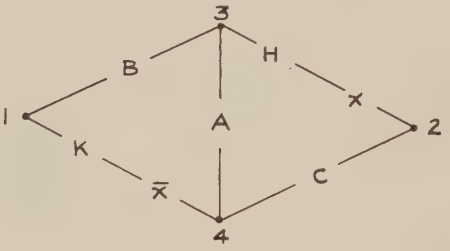
*Form 2:* The function contains clauses of the type

$$f = Ax(BC \vee -) \vee B\bar{x} \vee CH.$$

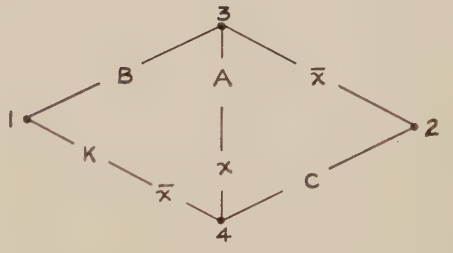
Joining the zero value clause $Ax\bar{x}H$ to $f$ gives

$$f = Ax(BC \vee \bar{x}H) \vee B\bar{x} \vee CH.$$

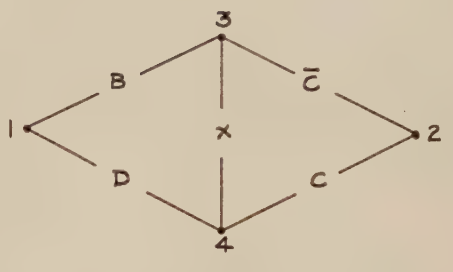
A resulting bridge network realization is seen in Fig. 15(b).




(a)




(b)




(c)

Fig. 15—Several bridge network forms resulting from redundancies.

*Form 3:* Let

$$f = Bx \vee Dx \vee B\bar{C} \vee CD$$

$$= x(B \vee D) \vee B\bar{C} \vee CD.$$

By the Type 3 modification, replace $Bx$ with $BCx$ and $Dx$ with $D\bar{C}x$. Then,

$$f = x(BC \vee D\bar{C}) \vee B\bar{C} \vee CD.$$

A resulting bridge network realization is shown in Fig. 15(c). Forms 1 and 2 use the Type 1 modification. These redundant clauses correspond to "blocked paths" in the network. This blocking serves to prevent "sneak paths." Form 3 uses no "blocked paths."

The following extension of Theorem 6 can be shown to hold.

*Theorem 8:* If a minimum $\vee$ polynomial representation of $f$ may be altered using redundancies in such a way that $f$ may be fully assembled as

$$f = g_1 g_2 \cdots g_k,$$

where $g_i$ is an $\vee$ polynomial, and no fully assembled representation exists without using all of the redundancies applied, then all the $g_i$'s are minimum $\vee$ polynomials.

If no further network synthesis can be accomplished using only redundancies which appear in the minimum $\vee$ polynomial representation, $M$, then redundant clauses using literals which do not appear in $M$ may be tried. An upper bound on the number of different literals that should be used simultaneously is the difference between the number of elements required for the simplest known series-parallel network realization and the number of different literals appearing in $M$. The synthesis procedures using such redundancies are the same as previously described.

Even though the synthesis procedure may lead to a simplest network realization, this in no way implies that this is a unique simplest network realization. In fact, the process was shown to yield a simplest realization only under very restricted conditions.

The three main advantages claimed for the procedure are first, a complex transmission function may be split into a set of less complex functions which can be realized more simply than the original complex function; second, the necessarily series and parallel parts of the network are recognized as such in the transmission function, and third, some simplest bridge network configurations are recognized in the transmission function with the aid of the bridge condition.

## VI. Conclusions

The analysis of two-terminal networks is carried out by a decomposition procedure described in Section IV. This decomposition procedure is also restated and used for network synthesis for the necessarily series or parallel parts of a network as determined from the transmission function for the network. A bridge condition also is developed to aid in the synthesis of two-terminal bridge networks. The minimum $\vee$ polynomial is used as a starting functional representation for synthesis. In addition, the use of redundant variables is considered as an aid to synthesis.

Under certain conditions the synthesis procedure is shown to yield network realizations with the fewest possible number of elements.

Some extensions to multiterminal networks are possible.[15] More complex conditions seem to be required, however, before adequate synthesis procedures can be described to give minimal network realizations for multiterminal networks. A counting and classification procedure for all bridge network types with $n$ vertices, for small $n$, would also find ready application to aid in the formalization of bridge network synthesis.

[15] Miller, *op. cit.*, pp. 71–78.

# A Transistor Pulse Generator for Digital Systems*

DOUGLAS J. HAMILTON†

*Summary*—**A design procedure is developed for a new transistor pulse generator circuit suitable for use as a building block in a digital system. The circuit produces a pulse whose shape is relatively independent of variations in transistor parameters and load current. Pulse durations in the range 0.5 microsecond to 20 microseconds and load currents of several hundred milliamperes may be obtained.**

## Introduction

THE requirements placed upon a pulse generator which is to be used as a building block in a digital computer or other pulse system may be generalized as follows:

1) The circuit is to be triggered by a waveform of somewhat arbitrary shape and repetition rate.
2) The circuit must provide an output pulse whose amplitude and shape are relatively independent of the input waveform, and are also independent of load variations over a considerable range.
3) Because the circuit is to be used repetitively throughout the system, the output pulse must be reasonably independent of variations in transistor parameters.

The blocking oscillator circuit appears attractive for such an application since it produces pulses whose shape is relatively independent of the input waveform. However, the circuit has several disadvantages which must be overcome if it is to be useful from a system point of view. From some fundamental considerations of transistor blocking oscillators a design procedure will be evolved for a new circuit which is capable of delivering pulses at repetition rates up to 500 kc, with durations in the range of 0.5 microsecond to over 20 microseconds having maximum variations in pulse duration of $\pm 10$ per cent due to variations of transistor alpha from 0.95 to 1.0 and load current from 20 ma to 200 ma.

* Manuscript received by the PGEC, May 1, 1958; revised manuscript received, June 18, 1958.
† Stanford Electronics Labs., Stanford, Calif.

## Some Fundamental Considerations of Transistor Blocking Oscillators

Two basic blocking oscillators circuit configurations, without bias or damping, are shown in Fig. 1. The circuit of Fig. 1(a) uses collector-to-base feedback and is referred to as "base control," while the circuit of Fig. 1(b) uses collector-to-emitter feedback and is referred to as "emitter control." A third configuration using emitter-to-base feedback is only slightly different from base control and is not discussed. For the purpose of analyzing the circuits the transformers are assumed to be ideal with a turns ratio $n$ and a primary inductance $L$.

As far as rise-times are concerned, there is no essential difference between the two circuits. For a given natural frequency corresponding to a growing transient in the case of emitter control it can be shown that there is another transformer turns ratio which will produce the same natural frequency in the case of base control.[1] It will, however, become apparent that emitter control is to be preferred where variations in pulse duration must be kept to a minimum.

The model[2] which is used to represent the transistor in discussing the pulse duration has been chosen because it retains only the properties which are essential to the understanding of the operation of the circuit; it is shown, together with its volt-ampere characteristic, in Fig. 2. For switching applications the model is a reasonable representation of an alloy junction transistor and has the following properties:

1) The impedance between any two terminals is high when both junctions are reverse biased.
2) When both junctions are forward biased the voltage between any of its terminals is zero.
3) When both junctions are forward biased the impedance between any of its terminals is zero.
4) When the ratio of collector current to emitter current exceeds a value $A$, the collector junction becomes reverse biased and the impedance between collector and base and between collector and emitter is high.

It will be recognized that $A$ is the dc alpha of the transistor. If base current is the independent variable, as it is in the case of base control, application of Kirchhoff's Current Law to the model shows that the collector junction will be reverse biased when

$$\frac{I_c}{I_b} > \frac{A}{1-A}.$$

With the models for the circuit and the transistor thus defined, one is in a position to study the pulse duration. If in some manner the emitter junction is forward biased, regeneration will occur and will continue

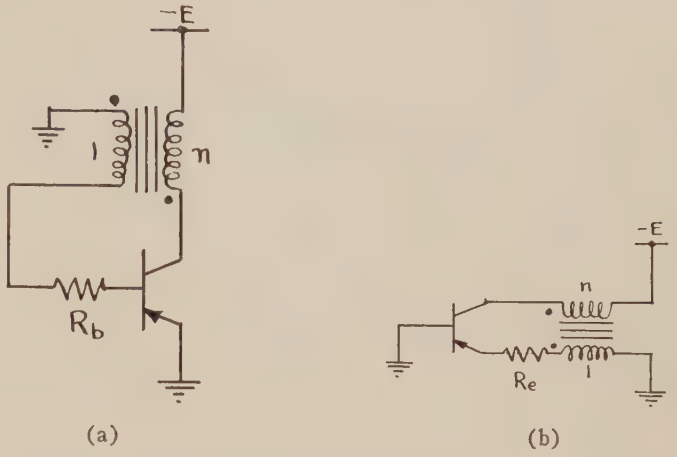



Fig. 1—Two basic blocking oscillator circuits. (a) Base control. (b) Emitter control.

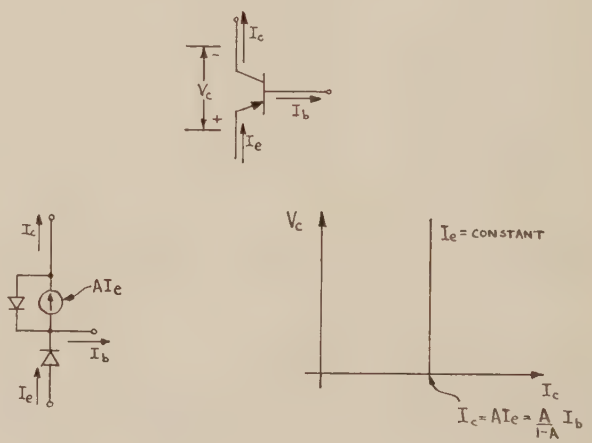



Fig. 2—Transistor model and its $V$-$I$ characteristic.

until the collector junction becomes forward biased; it will remain forward biased until the primary inductance of the transformer causes the collector current to increase to a value $I_c = AI_e$, at which time regeneration again occurs, continuing until both junctions are reverse biased. The pulse duration $\tau$ will be defined as the amount of time that the collector junction is forward biased.

Consider first the case of emitter control. For the duration of the pulse the collector junction is forward biased and the voltage across the primary winding of the transformer remains constant at $E$; hence the control current $I_e = E/nR_e$ flowing in the emitter circuit remains constant. The collector current, however, is made up of two components; one is $I_e/n$ and the other is $I_L$, the current in the primary inductance of the transformer. $I_L$ is initially zero but increases linearly with the time: $I_L = Et/L$. It is sometimes convenient to visualize this process in terms of required collector current $I_{cr}$ and maximum available collector current $I_{cmax}$,[3] as shown in Fig. 3. When the required collector current, which is $I_e/n + Et/L$, exceeds the maximum available collector

[1] J. G. Linvill, "Transistor electronics," class notes for course EE227, Stanford University, Stanford, Calif.; 1957.

[2] This is similar to a model proposed by R. B. Adler, "A large signal equivalent circuit for transistor static characteristics," Res. Lab. of Electronics, Mass. Inst. Tech., Cambridge; August 30, 1951.

[3] D. J. Hamilton, "Pulse Duration of Transistor Blocking Oscillators," Hughes Aircraft Co., Culver City, Calif., Rep. M-122; 1956.

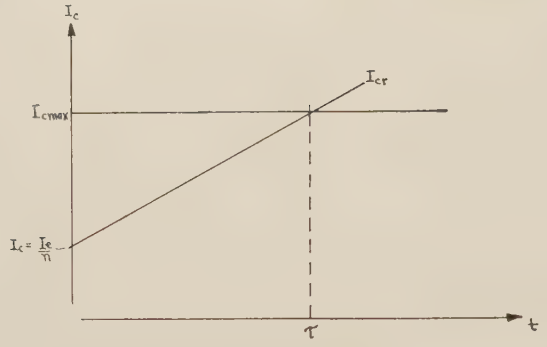



Fig. 3—Collector current during the pulse.

particular types of loading are shown in Fig. 4. If the load resistance is connected in shunt with the primary winding of the transformer as in Fig. 4(a), the required collector current is increased by an amount equal to the load current and a correspondingly shorter time is required for the collector current to increase to $I_c = AI_e$. If the load resistance is connected in series with the primary winding the voltage across the primary winding does not remain constant during the pulse, the current in the primary inductance does not increase linearly with time, and the emitter current does not remain constant. The resulting effect upon pulse duration is depicted in Fig. 4(b). The effects of other types of loading may be similarly deduced.

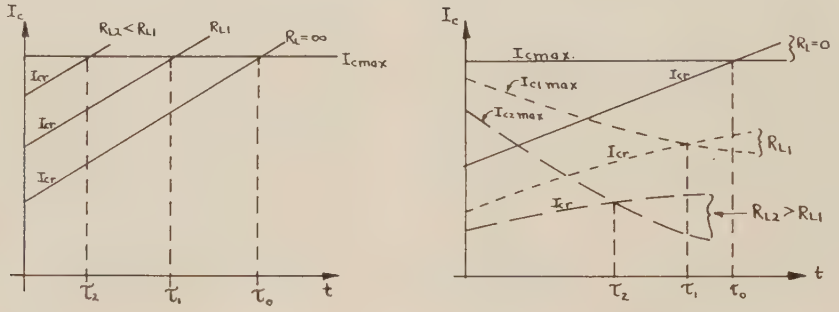



Fig. 4—Sketches showing the effect upon pulse duration of two types of loading. (a) $R_L$ shunting primary. (b) $R_L$ in series with primary.

current, which is $AI_e$, the collector junction becomes reverse biased and the pulse ends. The time thus elapsed is the pulse duration and is easily calculated to be

$$\tau_e = \frac{L}{nR_e}(A - 1/n). \tag{1}$$

Using similar reasoning for the case of base control one finds that

$$\tau_b = \frac{L}{nR_b}\left(\frac{A}{1 - A} - \frac{1}{n}\right). \tag{2}$$

Eqs. (1) and (2) show that to minimize variations in pulse duration due to changes in $A$, the transistor dc alpha, emitter control should be used in preference to base control since variations in $A$ will only be of the order of 5 per cent, while $A/1-A$ may change by 5 to 1 in a normally encountered distribution of transistors. Thus, further discussion is confined to emitter control.

There are several ways of connecting a load to the circuit, but in all cases a variation in load resistance produces a variation in pulse duration which would generally be intolerable in a system. It has been shown that loading the circuit also deteriorates the rise-time and the trigger sensitivity.[4]

Sketches of the effects upon pulse duration of two

It is thus apparent that if the circuit is to be useful in a system whose requirements are those previously stated, modifications must be made to couple the load to the circuit in such a way that the immittance presented to the circuit remains relatively constant with varying load.

## Limitations of the Model

The property of the model which led to an easy calculation of the pulse duration is that with both junctions forward biased the voltage between any of the terminals is zero until $I_c = AI_e$. In reality this is incorrect because of junction potentials and the presence of $r_b'$, the base spreading resistance. The latter is the major contributor to the voltage appearing between the base and collector and the base and emitter. If the model is modified to include $r_b'$, analysis reveals that the pulse duration is given by

$$\tau_e' = T \ln\left[1 + \frac{L}{nT}\frac{\left(A - \frac{1}{n}\right)}{[R_e + r_b'(1 - A)]}\right] \tag{3}$$

where

$$T = L\left[\frac{1}{r_b'} + \frac{1}{R_e}\left(\frac{1}{n} - 1\right)^2\right].$$

[4] J. G. Linvill and R. H. Mattson, "Junction transistor blocking oscillators," Proc. IRE, vol. 43, pp. 1632–1659; November, 1955.

As an example consider a case in which

$$L = 4.17 \text{ mh.}$$

$$n = 2$$

$$A = 0.98$$

$$R_e = 100 \text{ ohms}$$

$$r_b' = 100 \text{ ohms}$$

Eq. (3) yields $\tau_e' = 9.4$ microseconds. If $r_b'$ is assumed to be zero and (1) is used, one obtains $\tau_e = 10$ microseconds. In general, if $R_e \gg r_b'(1-A)$, and if accuracy of the order of 5 per cent is acceptable, (1) may be used in preference to (3).

Clearly, to make the pulse duration independent of transistor parameter variations one should make $R_e \gg \underline{r_b'}(1-\underline{A})$, where $\underline{r_b'}$ is the largest value of $r_b'$ that will be encountered and $\underline{A}$ is the smallest value of $A$ that will be encountered. This has the effect of maintaining the drop across $R_e$, and hence the emitter current, constant during the pulse.

Several 8 microsecond blocking oscillators with $R_e = 270$ ohms and Raytheon 2N113 transistors were constructed and in all cases the measured pulse duration was within 2 per cent of the value calculated by (1).

## A New Circuit

Two problems must be solved in order to produce from the blocking oscillator a circuit useful in a system. Bias circuitry must be provided so that the circuit is energized only by a triggering waveform, and some means must be devised to couple the load to the circuit so that variations in load will not alter the pulse duration and rise time. These problems are solved by a new circuit shown in Fig. 5.

In this circuit $T_1$ functions as a blocking oscillator, $T_2$ is used to couple the load to the circuit, and $D_1$ performs a dual function as a mechanism for providing cutoff bias and reverse base current to $T_2$.

The base-emitter circuit of $T_2$ is in the emitter loop of the blocking oscillator and all of the current flowing in the emitter of $T_1$ flows in the base of $T_2$. This current is made large enough to insure that the collector of $T_2$ is forward biased during the pulse under all conditions of loading, which insures that the amplitude will be constant. This method of coupling the load has several advantages. First, a large current is available in the collector circuit of $T_2$; second, the variation with load of the impedance between emitter and base of $T_2$ will be small and can be masked by $R_e$ so that the pulse duration is not affected; and finally, the impedance between emitter and base of $T_2$ is small enough to permit good rise-times to be obtained. A capacitor is placed in shunt with $R_e$ to eliminate the effects of $R_e$ during the rise time.

Proper bias voltages to insure that both transistors are cut off until the application of a trigger are provided by diodes $D_1$ and $D_2$, the former germanium, and the latter silicon. Both diodes are forward biased; $D_2$ by $(I_2 - I_1)$ and $D_1$ by $I_1$. Thus the emitter of $T_1$ is reverse biased by an amount equal to the voltage across $D_1$, and the emitter of $T_2$ is reverse biased by an amount equal to the difference in voltages across $D_1$ and $D_2$, normally about 0.4 volt. When a trigger is applied $D_1$ is reverse biased, $T_1$ turns on causing $T_2$ to turn on and regeneration occurs.

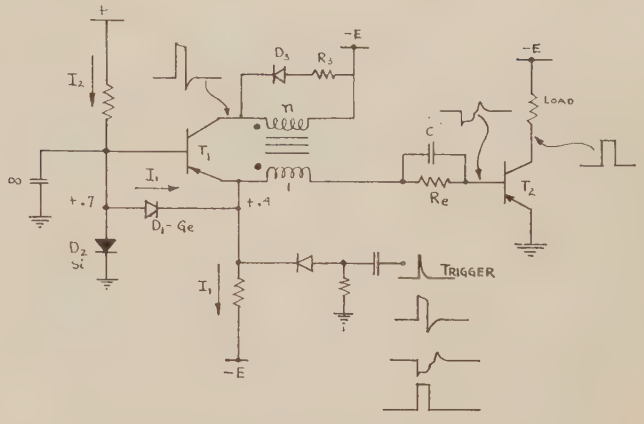



Fig. 5—New circuit which effects the desired improvements.

It is the second function performed by $D_1$ which makes the circuit practical. During the pulse the collector of $T_2$ is forward biased and a large number of minority carriers are stored in the base region. Assume for the moment that $D_1$ is not present. At the end of the pulse the voltage across the transformer reverses and the emitter of $T_1$ is driven negative. The base current of $T_2$ is reduced to zero and the minority carriers stored in the base are removed only by recombination. Because of the time required by this process, the duration of the pulse appearing across the load may be considerably longer than the pulse duration of the blocking oscillator. With $D_1$ in the circuit the emitter of $T_1$ is restricted to a reverse bias of only a few tenths of a volt. When the voltage across the transformer reverses, the base of $T_2$ is driven positive and reverse current flows, sweeping out the minority carriers and reducing the storage time to a negligible value. Hence the duration of the pulse across the load is the same as the pulse duration of the blocking oscillator. Because all of the energy stored in the transformer during the pulse may not be used in sweeping out the minority carriers, additional damping is provided by $D_3$ and $R_3$.

The ratio of reverse base current to forward base current is easily found. The forward base current $I_{bf}$ is equal to $I_e$, the emitter current of $T_1$. During the pulse the current built up in the primary inductance is $I_e(A - 1/n)$. When the pulse ends this current must continue to flow; it appears in the secondary winding multiplied by $n$ and in a direction opposite that of $I_e$. Hence, $I_{br}$, reverse base current, is $I_e(nA - 1)$. This assumes that $R_3/n^2$ is much larger than the impedance between emitter and base of $T_2$. The ratio of reverse to forward base current at the end of the pulse is therefore

$$\frac{I_{br}}{I_{bf}} = (nA - 1).$$

In calculating the pulse duration (1) must be modified to include the effect of bias voltage and the base-emitter circuit of $T_2$:

$$\tau_e'' = \frac{L}{nR_e}\left(A - \frac{1}{n}\right)\left[\frac{E - n(V_{be} + E_s)}{E + E_s}\right] \qquad (4)$$

where $V_{be}$, $E_s$, and $E$ are the magnitudes of the forward base-emitter voltage of $T_2$, the forward voltage of the silicon diode, and the supply voltage, respectively.

## Design Procedure

A design procedure may now be established to be used in adapting the circuit to a particular system:

1) Choose a value of base current $I_{bf}$ for $T_2$ large enough to insure that the collector will be forward biased during the pulse under all load conditions.
2) Select $R_e$ so that with this current flowing the drop across $R_e$ will be large compared to *variations* in the emitter-base voltage of $T_2$ which are anticipated under all conditions of load and transistor parameter variations. This amounts to making $R_e$ large enough to mask the variations in impedance between emitter and base of $T_2$ so that $I_e$, and hence the pulse duration, remains constant.
3) Select a value for $C$. The time constant of the emitter circuit of the blocking oscillator is approximately

$$\frac{R_e r_b' C}{R_e + r_b'}$$

where $r_b'$ is the base resistance of $T_2$. $C$ should be chosen so that this is about one fourth the pulse duration. Thus the pulse duration will not be influenced by the charging of the capacitor.
4) Choose a turns ratio $n$. Here one wishes to satisfy two criteria. When $n$ and $R_e$ have been selected a supply voltage is automatically determined: $E \cong nI_{bf}R_e$. It is necessary that $n$ be chosen so that the maximum collector voltage is not exceeded, particularly at the end of the pulse when the transformer voltage reverses. Within this restriction one wishes to select, if possible, a value of $n$ to obtain the best rise time. The selection of an optimum $n$ will be discussed in the next section.
5) Using (4) and the nominal value of $V_{be}$, select a value of primary inductance to produce the desired pulse duration.

The 0.5 microsecond pulse generator shown in Fig. 6 was designed to supply a load current of 20 ma to 200 ma. Two transistors are used in the output to minimize the effects of a decrease in transistor current gain at high emitter currents. The use of two transistors has the additional advantage of reducing the parasitic impedance in the emitter loop of the blocking oscillator. As

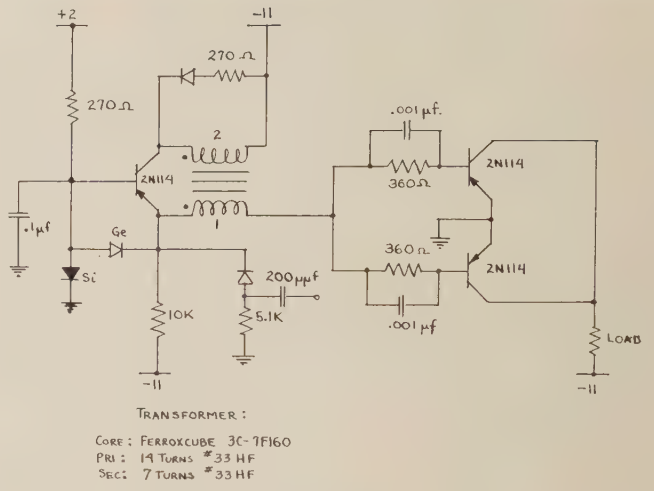



Fig. 6—Pulse generator for 0.5 microsecond, 250 kc operation.

can be seen from the waveforms in Fig. 7, the variation in pulse duration over the entire load range is less than 10 per cent. The circuit has been operated at pulse repetition frequencies as high as 500 kc.

In the tested circuits a 4-volt trigger was required for consistent operation, and no change in output rise-time was observed for triggers in excess of 8 volts.

## Choosing a Turns Ratio

The problem of choosing a turns ratio for the pulse generator which is to be used in a system can generally be reduced to the following terms:

Given the maximum $r_b'$ and collector capacitance $C_c$, and the minimum small-signal current gain $\alpha_0$ and radian cutoff frequency $\omega_0$, which are to be expected from the transistors which will be available, approximately what turns ratio should be used to obtain the minimum rise-time?

Using the method of Linvill and Mattson[4] an analysis will be given here for the circuit of Fig. 5 which expresses the optimum turns ratio in terms of the product $r_b'C_c\omega_0$.

The circuit of Fig. 5 has certain natural frequencies which correspond to growing transients, and the regenerative process may be thought of as a multiplication by these transients of the initial conditions established by a trigger source or other means. It is generally not necessary to calculate the actual rise time; in fact, to do so would be difficult because of the nonlinearities present in the regenerative process. One is interested in minimizing the rise time; this implies maximizing the growing transients. A simple model will be used which readily lends itself to the maximizing of the growing transients.

If a transformer with good coupling is used, such as one with a ferramic cup-core, the leakage inductance may be neglected; the presence of leakage inductance changes the actual rise time but does not materially alter the optimum turns ratio.[5] During the transient

[5] J. G. Linvill, private communication.

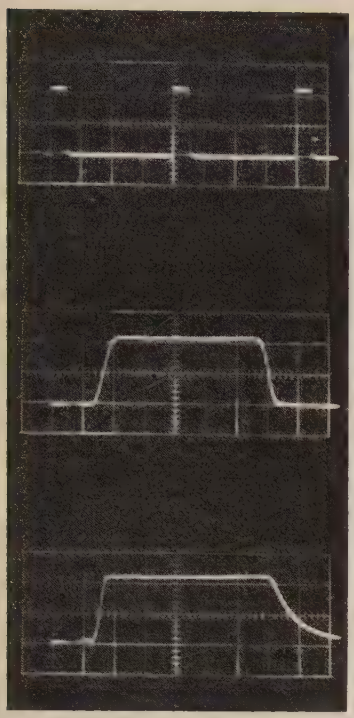



Fig. 7—Waveforms for the circuit of Fig. 6.

period the base-emitter impedance of $T_2$ consists primarily of $r_b'$, and the base-emitter circuit will be replaced by $r_b'$ in the model. The capacitor $C$ will usually be sufficiently large that its impedance may be neglected; this can later be verified by calculating its impedance at the natural frequency and comparing it with $r_b'$. The frequency dependence of alpha may be approximated by

$$\alpha = \frac{\alpha_0}{1 + \frac{s}{\omega_0}}$$

where $s$ is the complex frequency variable; and the model is then as shown in Fig. 8.

If one sets the determinant of the model equal to zero and solves for the natural frequency $S$ corresponding to a growing transient, the following is obtained:

$$r_b'C_c = \frac{\frac{1}{S}\left(\frac{n\alpha_0}{1 + S/\omega_0} - 1\right)}{2n^2 - 2n + 1}. \qquad (5)$$

It can be seen from (5) that for $\alpha_0$ in the range 0.95–1.0 the actual value of $\alpha_0$ will not have much influence upon $S$; therefore a further simplification can be made by assuming that $\alpha_0 = 1$. It is also more convenient to maximize $S/\omega_0$, the ratio of the natural frequency to the radian cutoff frequency. Letting $p = S/\omega_0$ one obtains

$$r_b'C_c\omega_0 = K = \frac{\frac{1}{p}\left(\frac{n}{1+p} - 1\right)}{2n^2 - 2n + 1}$$

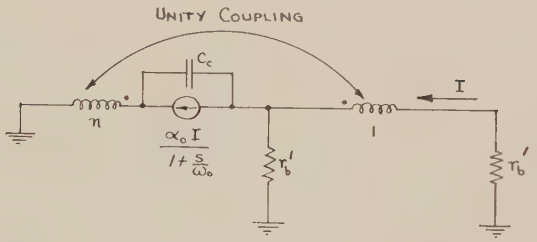



Fig. 8—Model of the circuit during the transient period.

from which

$$p = -\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4n}{(2n^2 - 2n + 1)K + 1}}.$$

The optimum value of $n$ to maximize $p$ is given by

$$n_{opt} = \frac{2}{3} + \frac{2}{3}\sqrt{1 + \frac{2}{3}\left(\frac{1+K}{K}\right)}. \qquad (6)$$

It will be noted that $K$ is the product of the three transistor parameters of interest: $r_b'C_c\omega_0$.

As an example, suppose that the available transistors have $r_b'_{max} = 100$ ohms, $C_{cmax} = 6.2 \times 10^{-12}$ farad, $\omega_{0min} = 62.8 \times 10^6$ radians per second ($f = 10$ mc). The optimum turns ratio is $n_{opt} = 3.55$, and $p_{max} = 1$ (the natural frequency is 10 mc). If $C = 0.001 \times 10^{-6}$ farad, the capacitive reactance at this frequency is 15 ohms.

It should be emphasized that the model used here is highly idealized and does not take into account the nonlinearities in emitter resistance and collector capacitance; the results are to be regarded as indicative only of the approximate turns ratio. It is also to be emphasized that the criterion that the maximum collector voltage of the transistor should not be exceeded must always be satisfied in choosing the turns ratio.

## Conclusions

The analysis in this paper has been aimed at presenting a simple picture which illuminates the operation and some of the shortcomings of the blocking oscillator. The analysis has led to a new circuit, suitable for use as a building block in a digital system, which eliminates the most troublesome of these shortcomings without resorting to undue complexity. The design procedure presented has been useful in adapting the circuit to several different systems. While the circuit manifestly cannot qualify as a precision pulse generator, the variations in pulse shape with load and transistor parameters are sufficiently small to permit its application in many systems whose requirements are not unduly stringent.

## Acknowledgment

Much of the original work on the circuits described was done at Hughes Aircraft Company, Culver City, Calif. The author is grateful to R. A. Day and J. W. Vorndran for helpful discussions during the course of this project, and also is indebted to Dr. J. G. Linvill of Stanford University for a critical reading of the manuscript.

# Corrections

Douglas B. Netherwood, author of "Logical Machine Design: A Selected Bibliography," which appeared on pages 155–178 in the June, 1958, issue of these Transactions, has called the following corrections to the attention of the editor.

On page 155, column 2, line 4, [23] should be read for [24]. On page 156, under Reading Suggestions, the references should be 29, 50, 61, 105, 143, 192, 234, 240, 262, 298, 361, 381, and 456. In the Bibliography itself, cross-references shown as "(400)" should read "[397]" for the following entries: [23], [108], [221], [283] [286], [440], [442], and [449].

---

D. W. Davies, author of "Switching Functions of Three Variables," which appeared on pages 265–275 of the December, 1957, issue of these Transactions, has requested the editor to publish the following corrected figure.

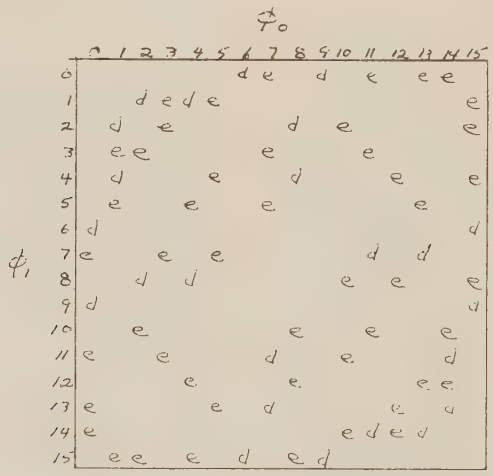



Fig. 5—Functions of classes *d* and *e*.

# Correspondence

## Switching Circuits as Topological Models in Discrete Probability Theory*

The purpose of this note is to call attention to a *substitution rule* which can facilitate discussion in the related areas of information theory, switching theory, computers, and probability theory. The discrete probability theory and its notation is usually confusing to the beginner. It is believed that the scheme to be suggested will facilitate both presentation and understanding of the subject.

The central notion in the probability theory is that of an event $E$ with a certain probability of occurrence $Q(E)$ or nonoccurrence $P(E)=1-Q(E)$. The event $E$ can be assigned a switching variable $e$ which takes the value 1 when the event occurs and 0 when it does not occur. A joint event $E$ involving two simple events $E_1$ and $E_2$ can be related to the simple events in the most general way by writing the logical connection between these events in the canonic form of a switching function of two variables[1]

$$e = f_0e_1'e_2' + f_1e_1'e_2 + f_2e_1e_2' + f_3e_1e_2 \quad (1)$$

where the coefficients $f_i$ can be either zero or one in the switching function. The primed variables are logical negatives of the unprimed variables, thus $e_1'=1$ is interpreted to mean that event $E_1$ did not occur. The sample space and events in that space may be interpreted effectively in terms of the topological model given for switching functions.[2] Simple and joint events correspond to collections of cells of various orders.

Because of the correspondence between switching functions and switching circuits, these circuits may serve as topological models of an experiment or system. As an example of such a model, the joint event $E$ which occurs if and only if $E_1$ and $E_2$ occur is represented by

$$e = e_1e_2$$

which, in turn, is represented by a switching circuit consisting, for example, of two independent relay contacts in series. In this representation, unity corresponds to closure and zero corresponds to an open contact or open network. Similarly an event $E$ which occurs if either or both of two events $E_1$ and $E_2$ occurs is represented by

$$e = e_1 + e_2$$

which, in turn, corresponds to the parallel connection of two independent relay contacts.

The fundamental relation which links the switching circuit model with the discrete probability theory is the following *substitution rule* which is offered without proof:

The probability $Q(E)$ of occurrence of a complex event which can be represented in terms of a switching function of $n$ variables in the complete product form

$$e = \sum_0^M f_ip_i \qquad (M = 2^n - 1) \quad (2)$$

where $p_i$ is a complete product, can be written from (2) by the following substitutions:

1) An unprimed variable is replaced by the probability $Q$ of occurrence of the associated event.

2) A primed variable is replaced by the probability $P=1-Q$ of non-occurrence of the associated event.

3) The Boolean $+$ and $\cdot$ go over into identical symbols in ordinary algebra.

As an example of the application of the substitution rule, consider the following experiment. Five numbered dice are thrown and a payoff $E$ occurs if sixes appear on one or more of the following combinations of the dice

(1, 2) (1, 4, 5) (3, 4) (2, 3, 5)

What is the probability of a payoff?

The switching function is

$$e = e_1e_2 + e_1e_4e_5 + e_3e_4 + e_2e_3e_5. \quad (3)$$

Two possible switching circuits which yield (3) are shown in Fig. 1. In the series-parallel network the branches are not independent, while in the bridge network the branches are independent. Whether the branches are independent or not, the substitution rule applies.

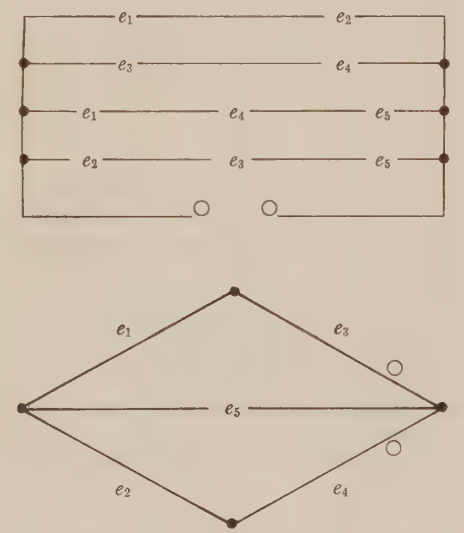



Fig. 1—Switching circuit models of experiment.

The canonic expansion of (3) yields

$$e=e_1e_2e_3e_4e_5+e_1e_2e_3e_4e_5'+e_1e_2e_3e_4'e_5+e_1e_2e_3e_4'e_5$$
$$+e_1e_2e_3'e_4e_5+e_1e_2e_3'e_4e_5'+e_1e_2e_3'e_4'e_5'+e_1e_2e_3'e_4'e_5$$
$$+e_1e_2'e_3e_4e_5+e_1'e_2e_3e_4e_5+e_1'e_2'e_3e_4e_5+e_1e_2'e_3'e_4e_5$$
$$+e_1'e_2e_3e_4'e_5+e_1'e_2e_3e_4e_5'+e_1'e_2'e_3e_4e_5'+e_1e_2'e_3e_4e_5'$$

and the substitution rule gives

$$Qe=Q_1Q_2Q_3Q_4Q_5+Q_1Q_2Q_3Q_4P_5+Q_1Q_2Q_3P_4Q_5$$
$$+Q_1Q_2Q_3P_4P_5+Q_1Q_2P_3Q_4Q_5+Q_1Q_2P_3Q_4P_5$$
$$+Q_1Q_2P_3P_4P_5+Q_1Q_2P_3P_4Q_5+Q_1P_2Q_3Q_4Q_5$$
$$+P_1Q_2Q_3Q_4Q_5+P_1P_2Q_3Q_4Q_5+Q_1P_2P_3Q_4Q_5$$
$$+P_1Q_2Q_3P_4Q_5+P_1Q_2Q_3Q_4P_5+P_1P_2Q_3Q_4P_5$$
$$+Q_1P_2Q_3Q_4P_5.$$

(Since each of the $Q$'s is $\frac{1}{6}$ and each of the $P$'s is $\frac{5}{6}$ in this example, the numerical result is that the probability of a pay off is 476/7776.)

In applying the switching circuit representation to the law of conditional probability, four possible probabilities will be considered:

$Q(A\,|\,B)$ which is the probability that $A=1$, given that $B=1$
$Q(A\,|\,B')$ which is the probability that $A=1$, given that $B'=1$ $(B=0)$
$P(A\,|\,B)$ which is the probability that $A=0$, given that $B=1$
$P(A\,|\,B')$ which is the probability that $A=0$, given that $B'=1$ $(B=0)$.

The last two may be written in terms of occurrence probabilities $Q$:

$$P(A\,|\,B) = Q(A'\,|\,B)$$
$$P(A\,|\,B') = Q(A'\,|\,B').$$

Hence only the occurrence probabilities are considered. The law of conditional probability states that for any $x$ and $y$

$$Q(x\,|\,y) = Q(xy)/Q(y)$$

and in this law $xy$ may be interpreted as a circuit consisting of networks representing $x$ and $y$ connected in series. Thus the probability of occurrence of $x$, when $y$ is known to have occurred, is the probability of closure of the network consisting of $x$ and $y$ in series, divided by the probability of closure of the network representing $y$. Thus for a conditional probability situation, there are *two* networks to which the substitution rule is applied.

As an example, each of the conditional probabilities is examined with

$$A = e_1 + e_2$$
$$B = e_2 + e_3.$$

Hence

$$AB = e_2 + e_1e_3 = e_1e_2e_3 + e_1e_2'e_3 + e_1'e_2e_3 + e_1e_2e_3' + e_1'e_2e_3'$$
$$AB' = e_1e_2'e_3'$$
$$A'B = e_1'e_2'e_3$$
$$A'B' = e_1'e_2'e_3'$$
$$A = e_1e_2e_3 + e_1e_2'e_3 + e_1'e_2e_3 + e_1e_2e_3' + e_1'e_2e_3 + e_1e_2'e_3'$$
$$B = e_1e_2e_3 + e_1e_2'e_3 + e_1'e_2e_3 + e_1e_2e_3' + e_1'e_2e_3' + e_1'e_2'e_3$$
$$B' = e_1e_2'e_3' + e_1'e_2'e_3'.$$

Then

* Received by the PGEC, March 31, 1958.

[1] M. Phister, "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y., pp. 50–53; 1958.

[2] R. H. Urbano and R. K. Mueller, "A topological method for the determination of minimal forms of a boolean function," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 126–131; September, 1956.

$$Q(A \mid B) = Q(AB)/Q(B) = \frac{Q_1Q_2Q_3 + Q_1P_2Q_3 + P_1Q_2Q_3 + Q_1Q_2P_3 + P_1Q_2P_3}{Q_1Q_2Q_3 + Q_1P_2Q_3 + P_1Q_2Q_3 + Q_1Q_2P_3 + P_1Q_2P_3 + P_1P_2Q_3}$$

$$= \frac{Q_1Q_3 + P_1Q_2 + Q_1Q_2P_3}{Q_3 + Q_2P_3}$$

$$Q(A \mid B') = Q(AB')/Q(B') = \frac{Q_1P_2P_3}{Q_1P_2P_3 + P_1P_2P_3} = Q_1$$

$$Q(A' \mid B) = Q(A'B)/Q(B) = \frac{P_1P_2Q_3}{1 - Q_1P_2P_3 - P_1P_2P_3} = \frac{P_1P_2Q_3}{1 - P_2P_3} = \frac{P_1P_2Q_3}{Q_1 + Q_2P_3}$$

$$Q(A' \mid B') = Q(A'B')/Q(B') = \frac{P_1P_2P_3}{Q_1P_2P_3 + P_1P_2P_3} = P_1.$$

It is not always desirable to use a canonic expansion and the substitution rule. When all switching elements in a network are independent, it is usually easier to write the probabilities from inspection of the network using series-parallel reduction. When the elements are not independent, the canonic expansion and substitution rule are simple, powerful tools.

J. N. Warfield
Purdue University
Lafayette, Ind

# Contributors

**Harvey L. Garner** (S'51–A'51–M'57) was born in Lake, Colo., on December 23, 1926. He received the B.S. and M.S. degrees from the University of Denver, Denver, Colo., and the Ph.D. degree from the University of Michigan, Ann Arbor, Mich.

From 1949 to 1951, he participated in the Cosmic Ray Research program at the University of Denver and the Inter-University High Altitude Research Laboratory, Echo Lake, Colo.

He was associated with the development and operation of the MIDSAC and MIDAC computers at the Engineering Research Institute of the University of Michigan during 1951–1955.

In September, 1955, he became an instructor in the Department of Electrical Engineering at the University of Michigan. In this capacity he has been active as a co-supervisor of the MIC computer project and has been in charge of several special intensive summer computer courses. He is presently assistant professor of electrical engineering.

Dr. Garner is a member of Sigma Xi, Sigma Pi Sigma, and the Association for Computing Machinery.

❖

**Roderick Gould** (S'55–M'58) was born in Shanghai, China, on December 25, 1932. He received the B.A. degree in mathematics from Amherst College, Amherst, Mass., in 1954, and the S.M. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, Mass., in 1955 and 1957, respectively. His graduate work was in the design and application of computing systems, and his doctoral dissertation dealt with the application of graph theory to contact network synthesis.

During 1957 and 1958, he was with the Comité d'Etude et d'Exploitation des Calculateurs Electroniques, Brussels, Belgium. Dr. Gould died earlier this year.

Dr. Gould was a member of the Association for Computing Machinery, the Society for Industrial and Applied Mathematics, the American Association for the Advancement of Science, Phi Beta Kappa, and Sigma Xi.

❖

**Harry J. Gray, Jr.** (S'45–A'46–M'55) was born in St. Louis, Mo., on June 24, 1924. He received the B.S. degree in electrical engineering in 1944, the M.S. degree in electrical engineering in 1947, and the Ph.D. degree in 1953, all from the University of Pennsylvania, Philadelphia.

After serving in the U. S. Navy as a radio specialist officer, he returned to the University of Pennsylvania in 1946. He worked on both the EDVAC and MSAC computers and then was associated with the development of a digital operational flight trainer (UDOFT). He contributed to the basic system organization of UDOFT, introduced the stability chart used to estimate the effect of accumulated truncation errors in the numerical integration logical packages of UDOFT, and to the peripheral equipment.

In 1954, he joined the Remington Rand Univac Division of Sperry-Rand Corporation, Philadelphia, Pa., where he worked on magnetic and transistor circuits. He developed the high-speed circuit system of LARC, and when he left in 1957 he held the position of staff engineer.

At present he is associate professor of electrical engineering in the Moore School of the University of Pennsylvania.

Dr. Gray is a member of the Association for Computing Machinery, Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

❖

**Douglas J. Hamilton** (A'53) was born in Canton, Ohio, on December 6, 1930. He attended the College of Wooster, Wooster, Ohio, and Case Institute of Technology, Cleveland, Ohio, receiving the B.S. degree from the latter institution in 1953. From 1953 to 1957 he was employed by Hughes Aircraft Co., Culver City, Calif., where he was engaged in the development of transistor circuitry for airborne digital computers. He received the M.S. degree from the University of California, Los Angeles, in 1956, and in 1957 joined the General Electric Company Computer Laboratory, Palo Alto, Calif., where he was engaged in the development of transistor circuitry for magnetic core and magnetic drum memories. Since January, 1958, he has been a research assistant at Stanford Electronics Laboratories, Stanford University, Stanford, Calif., where he is studying for the doctorate.

Mr. Hamilton is a member of Eta Kappa Nu, Tau Beta Pi, and an associate member of Sigma Xi.

❖

**Vincent J. Korkowski** was born on May 17, 1929, in Brandon, Minn. He graduated from Northwestern Television and Electronics Institute, Minneapolis, Minn., in 1954.

Since 1954, Mr. Korkowski has been with the Research Division of Remington Rand Univac, engaged primarily in research on magnetic storage devices and logical elements.

❖

**Charles A. Krause** (M'54) was born in Wahpeton, N. D., on October 18, 1926. He received the B.S. degree in electrical engineering from Iowa State College, Ames, in 1949 and the M.S. degree, also in electrical engineering, from Purdue University, Lafayette, Ind., in 1950. He entered the U. S. Army in 1944 and attended South Dakota State College, Brookings, S. D., and Rutgers University, New Brunswick, N. J., under the A.S.T.P. program. He was later in charge of a telephone repeater and shortwave radio station of the Alaska Communication System (Signal Corps).

From 1950 to 1953 he was a research engineer at Hughes Aircraft, Culver City, Calif, where he was engaged in the design, evaluation and prototyping of analog fire control system. From 1953 to 1955 he was a project engineer in the research department of the National Cash Register Company, Electronics Division, Hawthorne, Calif., where he performed technical coordination of a research and development program which included the application to digital systems of vacuum tubes, transistors, magnetic drums and heads, magnetic cores and ferro-electrics. He has been with the Norden Division, United Aircraft Corporation, Gardena, Calif., since 1955 and is currently head of the Special Equipment Section. His work there has included circuitry for machine tool control and for analog computing equipment, reliability studies, and the development of large-scale data handling systems.

Mr. Krause is a member of Tau Beta Pi, Phi Kappa Phi, Pi Mu Epsilon, Eta Kappa Nu and Sigma Xi.

❖

**Rodger R. Lowe** (SM'58) was born in Los Angeles, Calif., on December 26, 1926. He received the B.A. degree in theoretical physics from the University of California at Los Angeles in January, 1950.

From 1950 to 1953 he performed design and project engineering work in digital components and systems for the Electronic Engineering Company of California, Santa Ana, Calif. In 1954 and 1955 he served the National Cash Register Company, Electronics Division, Hawthorne, Calif., as research project engineer in advanced digital components and techniques.

From 1956 to the present time, Mr. Lowe has carried out special assignments in digital components and systems design and applications for the Norden Division, United Aircraft Corporation, Gardena, Calif.

❖

**Raymond E. Miller** (S'54—M'58) was born in Bay City, Mich., on October 9, 1928. He received the B.S. degree in mechanical engineering in 1950 from the University of Wisconsin in Madison. From August, 1950, to May, 1951, he was employed by International Business Machines Corporation in Endicott and Poughkeepsie, N. Y.

Until February, 1953, he served as a lieutenant with the U. S. Air Force. He then entered the University of Illinois Graduate College, and held a research assistantship in the Digital Computer Laboratory from June, 1953, to September, 1956, and a research associateship from June to September, 1957. He received the B.S. degree in electrical engineering in 1954, the M.S. degree in mathematics in 1955, and the Ph.D. degree in electrical engineering in 1957. He held the RCA Fellowship from September, 1956, to June, 1957.

At present, he is a staff engineer in the Information Research Department of the IBM Research Center, Yorktown Heights, N. Y.

Dr. Miller is a member of the Association for Computing Machinery, American Society of Mechanical Engineers, Pi Tau Sigma, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.

❖

**Joseph Otterman** (S'52–M'56) was born on April 12, 1925, in Warsaw, Poland. He attended the Hebrew Institute of Technology (Technion), Haifa, Israel, and was graduated in 1947 with the degree of Ingenieur in Electrical Engineering. During 1948–1950 he served in the Israeli Army, ending his military service as a lieutenant in the capacity of engineering reconnaissance officer of the Command of the South. From 1950 to 1951 he worked in the Israeli Government Scientific Institute on the development of instrumentation systems for ballistic measurements.

Coming to the University of Michigan, Ann Arbor, in 1951, he obtained the M.S.E. and Ph.D. degrees in electrical engineering in 1952 and 1955, respectively.

Since 1955, he has been associated with the Engineering Research Institute, University of Michigan, working on rocket investigation of the upper atmosphere within the International Geophysical Year program, and on problems of simulation by analog and digital techniques.

Dr. Otterman is a member of the American Physical Society and Sigma Xi.

❖

**William M. Overn** was born on March 20, 1924, in St. Paul, Minn. He served as a communications and radar officer with the U. S. Air Force in the Pacific area during World War II. In 1949 he received the B.E.E. degree in physics from the University of Minnesota, Minneapolis, where he did graduate work until 1950. From 1951 through 1952 he engaged in instrumentation work on the U. S. Corps of Engineers Underground Explosion test series as an employee of Engineering Research Associates, Inc.

Since 1953 Mr. Overn has been with the Research Division of Remington Rand Univac, engaged primarily in research on magnetic and ferroelectric storage devices and logical elements.

❖

**J. E. Robertson** (S'48–A'50–M'55) was born in Fairfax, Okla., on March 30, 1924. He received the B.S. degree in electrical engineering from Oklahoma A. and M. College, Stillwater, Okla., in 1947, the M.S. degree in 1948, and the Ph.D. degree in 1952, both from the University of Illinois.

He has been associated with the Digital Computer Laboratory of the University of Illinois since 1950, and is now a research associate professor of electrical engineering.

Dr. Robertson is a member of the Association for Computing Machinery.

❖

**Jack L. Rosenfeld** was born in Pittsburgh, Pa., on June 6, 1935. In 1957 he received the S.B. and S.M. degrees from the Massachusetts Institute of Technology, Cambridge, Mass., where he is now studying for the Sc.D. degree in electrical engineering.

He was employed by the Bell Telephone Laboratories as a cooperative course student, working on microwave tube development and electronic switching. He has also been associated with Texas Instruments, Incorporated, and Space Technology Laboratories.

Mr. Rosenfeld is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.

❖

**Thomas D. Rossing** (SM'56) was born in Madison, S. D., on March 27, 1929. He received the B.A. degree in physics and mathematics from Luther College, Decorah, Iowa, in 1950. Graduate study at Iowa State College, Ames, led to the M.S. degree in 1952 and the Ph.D. degree in physics in 1954. During this period he conducted research on ultrasonic dispersion in gases.

From 1954 to 1957, he was employed in the Research Division of Remington Rand Univac. His interests were in the field of ferromagnetic materials and devices. He has done research on devices for nondestructive readout and on vacuum-deposited ferromagnetic films. Since 1957, he has been an associate professor of physics at St. Olaf College, Northfield, Minn.

Dr. Rossing is a member of the American Physical Society and Sigma Pi Sigma.

JOINT COMPUTER COMMITTEE

# SENEWS

SCIENCE EDUCATION SUBCOMMITTEE NEWSLETTER

Vol. 1, No. 2 September, 1958

## Purpose

*SENEWS* is a newsletter addressed to computer oriented members of IRE, ACM, and AIEE to help them promote interest and knowledge among high school age students; it provides a nationwide medium of communication pinpointed to this subject; it evolves from volunteer efforts of the JCC Science Education Subcommittee and relies on its readers for news material.

Write to: C. W. Farr, Chairman, JCC Science Education Subcommittee, M.I.T. Lincoln Laboratory, Lexington 73, Mass.—or to your representative on the *SENEWS* Editorial Board: Richard W. Melville, IRE; George E. Forsythe, ACM; G. L. Hollander, AIEE.

## Dangerous Voltage

*SENEWS* bristles with stories of computer activity in high schools; but *do not* expect your local authorities to welcome you with open arms when you knock on the door and announce your intention to help them "discover" computers. Dr. L. C. Van Atta, a leader in the energetic and successful industry-education program in southern California, has written us: "A word of warning. It has been my experience that school systems tend to regard individual companies, specialized professional groups (petroleum engineers, aeronautical engineers, electrical engineers), and others as pressure groups unless their offers of assistance to the schools are on a very broad basis. Naturally the schools and community are interested in the total education of the student, rather than in biasing the student toward any particular specialization."

Dr. Van Atta went on to point out that innovations (no matter how good) represent a disruption to established curricula. Since Sputnik the well-meaning offers of assistance have multiplied. In Los Angeles the superintendent of schools has created a central committee for cooperation, and the school system has provided a full time Executive Secretary to coordinate the community efforts and "arrive at a broad program in which al specialized subjects have their appropriate emphasis."

Avoid burning up the circuit; measure the input impedance of the school system in your community before applying your driving voltage.

## Have Design, Will Build—Chapter Two

In the last issue we reported how David Ecklein, a high school junior from Cedar Falls, Iowa, was building a checker playing digital computer at home, using his own design, thirty-five hundred surplus tubes, and the help of his buddies in Tom Sawyer style. (*Ed. note:* Last minute correction of number of tubes was not in time for all publications.)

David's project has been interrupted for a while by summer employment at IBM's research laboratory in Poughkeepsie, N. Y., where he will learn to program and operate one of the major computers. He is looking forward to talking with engineers about circuit designs, and learning about the well-known checker playing program of Dr. A. L. Samuel. Nobody who has met 17-year old David will be greatly surprised if he makes some significant improvements in the techniques before the summer is out.

David Ecklein's opportunity to work in a major computer laboratory came about through his contact with the JCC Science Education Subcommittee. It is, in David's words, "beyond my fondest hopes. Not only will it provide experience but also a means to earn funds to carry out my project to conclusion."

Werner Buchholz

## Junior High School Computer Programmers

The thirteen year old in your home may not have learned to write the following coded program for the IBM 704 computer to solve (for 5 values each of $a$, $b$, and $c$) the equation $P=(a+b)(a^2+ab+b^2)(c+1)$. But don't be discouraged. We know of only one Junior High which has provided such instruction.

TABLE I

| Symbol | Operation | Address, Tag, Decrement | Remarks |
|---|---|---|---|
| SBM | LXA | *HERE, 1 | Load index register |
| | CLA | *A DATA+5, 1 | $(a+b)$ |
| | FAD | *B DATA+5, 1 | |
| | STO | *TEMP | |
| | LDQ | A DATA+5, 1 | $a^2$ |
| | FMP | A DATA+5, 1 | |
| | STO | TEMP+1 | |
| | LDQ | A DATA+5, 1 | $ab$ |
| | FMP | B DATA+5, 1 | |
| | STO | TEMP+2 | |
| | LDQ | B DATA+5, 1 | $b^2$ |
| | FMP | B DATA+5, 1 | |
| | STO | TEMP+3 | |
| | CLA | TEMP+1 | $(a^2+ab+b^2)$ |
| | FAD | TEMP+2 | |
| | FAD | TEMP+3 | |
| | STO | TEMP+4 | |
| | CLA | *C DATA+5, 1 | $(c+1)$ |
| | FAD | *INFO | |
| | STO | TEMP+5 | |
| | LDQ | TEMP | $(a+b)(a^2+ab+b^2)=$(PROD.) |
| | FMP | TEMP+4 | |
| | STO | TEMP+6 | |
| | LDQ | TEMP+5 | (PROD.)$(c+1)=$(ANS.) |
| | FMP | TEMP+6 | |
| | STO | *X INFO+5, 1 | |
| | TIX | SBM+1, 1, 1 | If contents of index register are greater than decrement (1), decrease contents by decrement (1) and transfer to SBM+1. Otherwise proceed to next instruction (HTR) |
| | HTR | SBM | Halt and return to starting position. |

* Note: 5 is in location 5 HERE; 5 values of "$a$" start at location A DATA, 5 $b$'s start at B DATA, 5 $c$'s start at C DATA: number 1 is in location INFO; TEMP is a temporary storage register; X INFO is storage location of final result.

Harley Tillitt of the U. S. Naval Ordnance Test Station, China Lake, Calif., conducted an educational experiment with 24 selected eighth grade students at Burroughs Junior High School in November, 1957. The experiment involved lectures and demonstrations during 10 forty-minute instruction periods. Computer programmers will recognize the conventions developed by SHARE, an informal programming organization of IBM 704 users.

"The success of the students in these experiments shows that it is feasible to introduce computer programming instruction below the college level," concluded Tillitt in the report submitted to *SENEWS*. The problem coded here (Table I) is one of 14 problems assigned to eighth grade students during the experiment. The complete report came to *SENEWS* via George Forsythe. (*Ed. note:* If this report is published in more detail, *SENEWS* will announce it in a later issue.)

## SADSAC II

Samson Additive Digital Sequential Automatic Computer, SADSAC II, carried its designer-builder, Peter Samson, to first place in the Massachusetts State Science Fair, and fourth place in the 1958 National Science Fair. Peter lives in Lowell, Mass., and will enter M.I.T. this autumn with a four year National Merit Award scholarship. He visited Lincoln Laboratory recently on invitation from the Science Education Subcommittee—and when he left it was *we* who were inspired.

First we reviewed specifications. SADSAC II is a relay computer, designed to add, subtract, and complement four bit binary numbers; input is from two punched paper tape readers, and programming provides for jumping from one tape to the other; output is a solenoid actuated typewriter (actually a home rigged 1898 Oliver).

But then we explored motivation, and the "engineering economics" of student computers. Peter's early science interest found roots in his father's electrical (hi-fi, etc.) equipment. High school algebra whetted his mathematics appetite. Attempting to build a machine to play ticktacktoe led him to design his own arithmetic circuitry, and the home grown SADSAC I computer was the inevitable result. After winning local honors with SADSAC I he went on to national fame with its successor.

Peter did not have help from a computer expert advisor; like most student builders of computers he dug the know-how out for himself, "and used common sense," he hastens to add. He regarded pertinent literature as scarce but good; the hardware situation he found pretty sad. We asked why he used relays instead of tubes or transistors. He pointed out that relays are cheap and are ac energized at a few standardized voltages; tubes and transistors involve not only more money, but more complex circuitry and power supplies. He used discarded pinball machine components, and reliability was a nightmare.

"Building SADSAC III this summer, Peter?"

"No. I want to get in some study that I can't find time for during the school year."

## Book and Film

"Local action kit," prepared by the President's Committee on Scientists and Engineers, Washington 25, D.C.

This kit is designed to: 1) tell how to bring together community organizations with the same basic objective into a single program, 2) promote self-evaluation and the recording of successful techniques and experiences useful to other communities throughout the nation, and 3) establish a pattern of nationwide effort without sacrificing the incentives and benefits of creative local action.

Here is what the kit contains: 1) "Guidebook for local action," suggesting general techniques adaptable

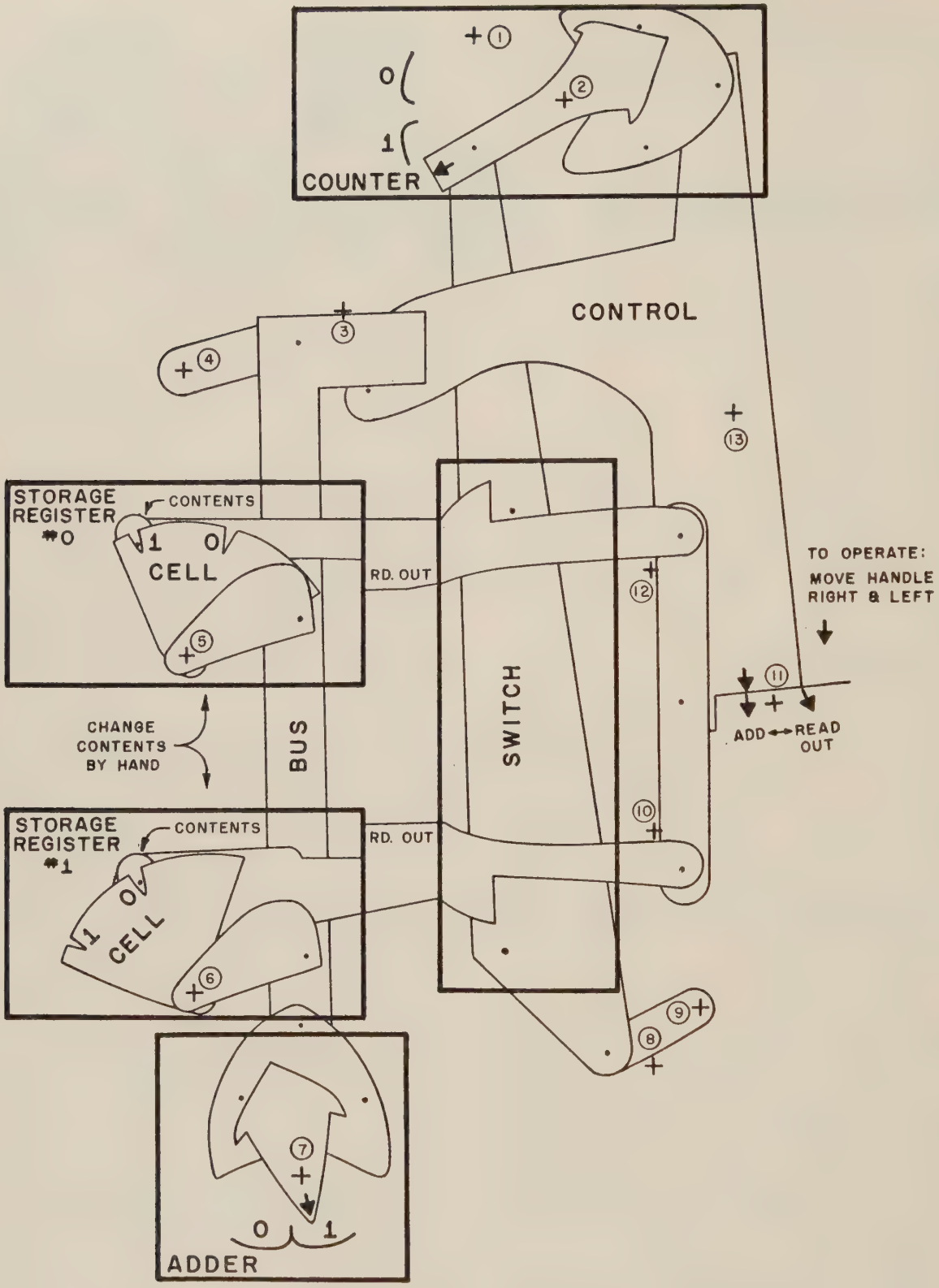



Fig. 1—PAPAC-00, a 2-register, 1-bit, fixed-instruction binary digital computer (Rollin P. Mayer, July 14, 1958).

to the needs and resources of your own community, 2) examples of tested projects in other communities, relating how they began, how they are being carried out, their scope, financing, and results, 3) reference materials such as information on scholarships, improvement of science curricula, youth activities, etc., and 4) a bibliography of selected materials, visual aids, organizations, and publications helpful in your community program.

This kit was prepared for use in local programs for the improvement of science and mathematics education (not limited to computer education). It is available without charge to *groups*, not (for budgetary reasons) to individuals.

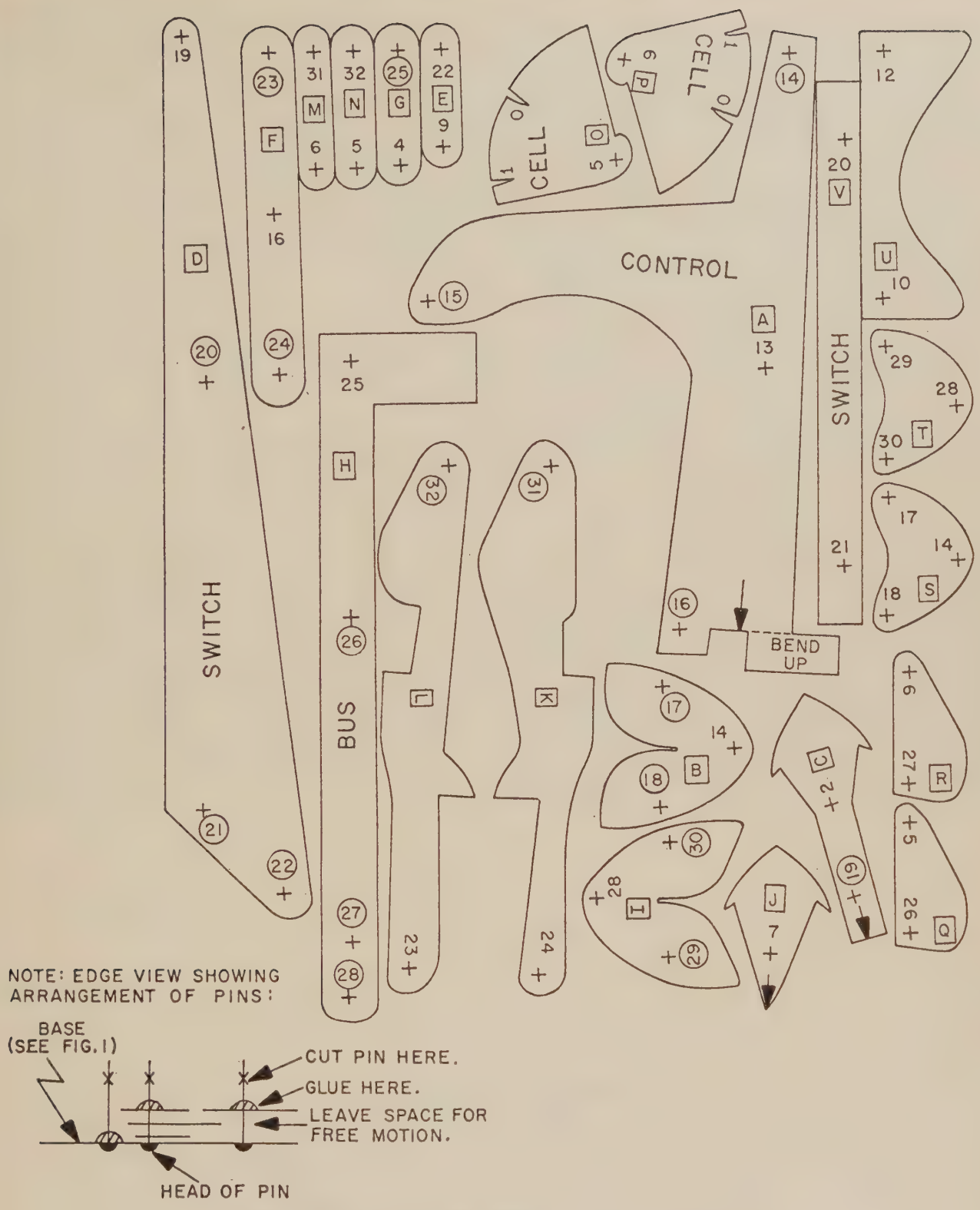



Fig. 2—Parts for PAPAC-00 (Rollin P. Mayer, July 14, 1958).

Interested individuals can get free copies of "Local ction," the monthly news bulletin. (*Ed. note:* Don't e misled by the price; this is good stuff.)

## PAPAC-00, A Do-It-Yourself Paper Computer

In less than an hour you can build the simplified gital computer shown in Fig. 1, using only a pair of issors, three dozen common pins, and the parts shown in Figs. 1 and 2. This computer was developed from the model demonstrated in the Concord, Mass., High School lectures on computers reported in the first issue of *SENEWS*.[1]

From the discussion below, the computer expert will

[1] *SENEWS*, IRE Trans. on Electronic Computers, vol. EC-7, pp. 186–187; June, 1958.

recognize that "PAPAC double zero" contains most of the units of a large-scale computer, but in simplified form. The *control unit* includes a counter and a system for controlling the parts of the computer according to the instruction being performed (in this model a simple fixed instruction is used; a large computer can draw from several instructions obtained from storage). The *storage unit* includes registers, bus, and selection switch; register contents are changed by hand rather than by the computer. The *arithmetic unit* can add. *Input and output units* have been eliminated by allowing the operator to deal with the insides of the computer directly rather than by way of complicated equipment. Proprietary rights are held by the author.

In *operation*, PAPAC-00 follows the same fixed instruction over and over again. This instruction is: "Read the number out of the currently-selected storage register and add it to the adder, then get ready to use the next storage register for the next time." The "counter" keeps track of which storage register to use next; since there are only two registers, numbered "0" and "1," the counter alternates between them. The "switch" is controlled by the counter and allows only the selected register to be operated. Each "storage register" contains only a single binary "cell"; when the register is operated, the cell is forced against the "bus" if the cell is set to "1." If a "1" has been read out in this way, the bus actuates the "adder," preparing it to add the "1." If the cell is set to "0," the bus and adder are not operated, and "0" is added to the adder. Binary sums are as follows: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. The adder forms these sums correctly except that in the last case it forms a sum of "0" because it can handle only one digit The "control," pushed back and forth by hand, performs this fixed instruction by operating the counter and switch, and by returning the bus to its "0" position (if it had read out a "1") causing the sum to be formed in the adder.

To *assemble* PAPAC-00, Fig. 1 should be used as the base, and the shapes of Fig. 2 should be fitted over it by following these steps:

1) Punch a pinhole exactly through the intersection of each cross (+) in Figs. 1 and 2 (but not the dots in Fig 1).

2) Cut out exactly on the lines, the parts in Fig. 2, in any order They are marked with a letter in a square box, from [A] to [V], and the next steps will be easier if you place each piece on the table in alphabetic order as you cut it out.

3) Place a pin up through each hole with a circled number (from (1) to (32)).

4) Taking each part of Fig. 2 in alphabetic order, place its *un*circled number holes down over the correspondingly numbered pins.

5) In first operating the computer you may find that some parts jam because the upper piece is down too far on the pins: pry such pieces up a little to provide space for free motion.

6) The construction can be refined by cutting the pins and gluing the uppermost part to the remaining length. Caution:

1) Don't cut the stop pins too short.

2) Glue only *one* moving part to the same pin.

ROLLIN P. MAYER

# PGEC News

## 1959 Western Joint Computer Conference

Papers are being solicited for the 1959 Western Joint Computer Conference, to be held at the Fairmont Hotel, San Francisco, Calif., on March 3–5, 1959. The theme of this conference will be "New Horizons with Computer Technology."

In keeping with the theme there is particular emphasis on factual papers dealing with the newer applications of computer techniques, such as Information Retrieval, Operation Control, Pattern Analysis, Decision Making, Computer Communications, Learning Concepts, and so forth, as well as on papers dealing with advances in computer component and systems design.

It is also hoped that there will be two sessions of a speculative nature: A "Blue Sky Session" and "Philosophy and Responsibility of Computers in Society." Papers intended for the "Blue Sky Session" should deal with the extension of computer technology into areas not considered feasible at present. They should indicate the advantages of such extension, and also the area of research necessary to bring this application into the feasible range. Papers for the session on "Philosophy and Responsibility of Computers in Society" should deal with the philosophic and/or social implications of the widespread application of automatic computer techniques. The papers for these sessions should be of the type to invite serious discussion. These two sessions will be definitely scheduled only if enough suitable papers are received.

Papers for the 1959 WJCC should be prepared based on a 20-minute delivery time. Selection of papers for presentation will be made from the complete text of the paper. There are no format requirements for these submission drafts. Three copies of the proposed paper should be submitted to Murray L. Lesser of the Technical Program Committee by October 1, 1958, and addressed to Mr. Lesser at IBM Research Laboratory, San Jose, Calif. After review, final selection of papers will be made and the authors will be notified by December 1. Submission of final texts of the selected papers, in the form required by the Publications Committee, should be made by February 1, 1959.

The 1959 WJCC makes the classical promise to be an exciting conference. Whether or not you are in a position to present a paper, you will find it well worthwhile to reserve March 3–5, 1959, to attend.

## 1958 Western Joint Computer Conference

On May 6 through 8 the very successful 1958 Western Joint Computer Conference, "Contrasts in Computers," was held at the Ambassador Hotel, Los Angeles, Calif. Regarded by the Conference Committee as the "150 per cent Conference" because every event—technical as well as social—was about half again as large as predicted, the Conference registered a total of over 1800 people. It was the largest of the Western conferences held so far. An outstanding program was presented by the Technical Program Committee. In the keynote session, which was devoted to exploring the possible social problems which may arise from widespread automation, a number of very intriguing and significantly new computer applications in the social and behavioral sciences were suggested. Six invited panel discussions presented and explored several of the controversial questions of the present computer art, and six sessions of contributed papers described some of the recent advances in computer hardware and computer applications. A "bonus feature" of the technical program was the added technical session devoted to a paper describing the recently announced French Gamma-60 computer.

Fifty exhibitors demonstrated recent components, equipment, techniques, and company progress in the 80 booths of the exhibit area. The social events of the Conference—the women's activities, the cocktail party and the luncheon—were all over-subscribed, and contributed to the over-all conference success and atmosphere. Following the Conference a one-day Symposium on "Small Automatic Computers and Input /Output Equipment" was sponsored by the Los Angeles Chapter of the ACM, and attended by 350 people. Eight papers from the manufacturers described their latest equipment.

Proceedings of the Conference will be published in the late fall of 1958 and will be available to nonregistrants from the three co-sponsoring societies: the IRE, the ACM, and the AIEE. A Proceedings of the Symposium will be published at a later date by the ACM.

### Call for Papers

1959 IRE National Convention

**March 23–26, 1959**

**Waldorf-Astoria Hotel and New York Coliseum, New York, N. Y.**

Prospective authors are requested to submit all of the following information by:

**October 24, 1958**

1. 100-word *abstract in triplicate*, title of paper, name and address
2. 500-word *summary in triplicate*, title of paper, name and address
3. An indication of the technical field in which the paper falls (*e.g.*, Electronic Computers).

Note: Only considered are original papers not published or presented prior to the 1959 IRE National Convention; any necessary military or company clearance of paper must be granted prior to submittal.

Address all material to: Dr. George L. Haller, Chairman
1959 Technical Program Committee
The Institute of Radio Engineers, Inc.
1 East 79 Street, New York 21, N. Y.

## INFORMATION FOR AUTHORS

The PGEC TRANSACTIONS is published quarterly and will bear datelines of March, June, September, and December. Abstracts of papers appearing in the TRANSACTIONS will appear also in IRE PROCEEDINGS. The PGEC publication schedule requires about one month for review and correction of all accepted manuscripts. The professional IRE Editorial Staff requires an additional two months' production time from receipt of manuscripts to completion of the printed journal.

**MANUSCRIPTS:** Three copies of the manuscript should be submitted. They should be typewritten (original and two carbon copies), and double spaced on only one side of each sheet. References should appear as footnotes, numbered consecutively, and include in the following order the author's name (including initials), title of reference work, journal name, volume, initial and final page numbers, and date of publication. Footnotes should be listed on a separate sheet and not inserted in text. Each paper must be accompanied by two copies of a summary not more than 200 words in length. Reviewing normally will require about four weeks from receipt of manuscript.

**ILLUSTRATIONS:** Only original illustrations should be submitted; they will be returned if desired. Photostatic copies of originals are not acceptable, except where they are exceptionally clear, with sharp black and white contrasts. All line drawings (graphs, charts, diagrams, etc.) should be prepared on drafting cloth or white drawing paper in India ink. It is preferable that only the coordinate lines show in graphs. All lettering must be large enough to be legible when reduced 50 to 75 per cent in size. Photographs should be glossy prints. Figure numbers should be indicated on the back of each illustration. Figure numbers and captions should be listed on a separate sheet accompanying manuscript. All drawings, photographs, and other manuscript material should be not larger than $8\frac{1}{2}$ by 11 inches for ease in handling.

Please submit all manuscripts to

J. P. Nash, PGEC Editor,

Missile Systems Division, Lockheed Aircraft Corp.,

3251 Hanover Street, Palo Alto, California.